digital

OpenVMS AXP Version 6.1
New Features Manual

OpenVMS

Part Number: AA-Q2HNA-TE

OpenVMS AXP Version 6.1 New Features Manual

Order Number: AA–Q2HNA–TE

March 1994

This manual describes the new features of the OpenVMS AXP Version 6.1 operating system.

Revision/Update Information:	This is a new manual.
Software Version:	OpenVMS AXP Version 6.1

Digital Equipment Corporation
Maynard, Massachusetts

March 1994

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

© Digital Equipment Corporation 1994. All rights reserved.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation: Alpha AXP, AXP, Bookreader, DECamds, DECevent, DECnet, DECdtm, DECwindows, Digital, LAT, MSCP, OpenVMS, OpenVMS RMS, OpenVMS Volume Shadowing, PATHWORKS, POLYCENTER, Q–bus, TMSCP, TURBOchannel, ULTRIX, VAX, VAX DOCUMENT, VMScluster, and the DIGITAL logo.

The following are third-party trademarks:

Adobe, Display PostScript, and PostScript are registered trademarks of Adobe Systems Incorporated.

IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers.

Internet is a registered trademark of Internet, Inc.

Motif and OSF are trademarks of the Open Software Foundation, Inc.

UNIX is a registered trademark of Unix Systems Laboratories, Inc., a wholly owned subsidiary of Novell, Inc.

All other trademarks and registered trademarks are the property of their respective holders.

ZK6317

This document is available on CD–ROM.

This document was prepared using VAX DOCUMENT Version 2.1.

Send Us Your Comments

We welcome your comments on this or any other OpenVMS manual. If you have suggestions for improving a particular section or find any errors, please indicate the title, order number, chapter, section, and page number (if available). We also welcome more general comments. Your input is valuable in improving future releases of our documentation.

You can send comments to us in the following ways:

- Internet electronic mail: OPENVMSDOC@ZKO.MTS.DEC.COM
- Fax: 603-881-0120 Attn: OpenVMS Documentation, ZKO3-4/U08
- A completed Reader's Comments form (postage paid, if mailed in the United States), or a letter, via the postal service. Two Reader's Comments forms are located at the back of each printed OpenVMS manual. Please send letters and forms to:

  Digital Equipment Corporation
  Information Design and Consulting
  OpenVMS Documentation
  110 Spit Brook Road, ZKO3-4/U08
  Nashua, NH 03062-2698
  USA

You may also use an online questionnaire to give us feedback. Print or edit the online file SYS$HELP:OPENVMSDOC_SURVEY.TXT. Send the completed online file by electronic mail to our Internet address, or send the completed hardcopy survey by fax or through the postal service.

Thank you.

Contents

Preface xiii

1 Overview of OpenVMS AXP Version 6.1 New Features

2 General Features

2.1 Common DCL Commands 2–1
2.1.1 Platform-Specific Attributes Have Been Removed 2–1
2.1.2 DEBUG Command 2–2
2.1.3 DIAGNOSE Command 2–3
2.1.4 DIFFERENCES Command 2–3
2.1.5 DIRECTORY Command 2–3
2.1.5.1 New Keywords for /SELECT Qualifier 2–3
2.1.5.2 /SHELVED_STATE Qualifier 2–3
2.1.5.3 Output Navigation Qualifiers 2–3
2.1.6 DUMP Command 2–3
2.1.7 F$FILE_ATTRIBUTES Lexical Function 2–4
2.1.8 F$GETDVI Lexical Function 2–4
2.1.9 HELP Command 2–4
2.1.10 PRODUCT Command 2–4
2.1.11 RECALL Command 2–4
2.1.12 REPLY Command 2–5
2.1.13 SEARCH Command 2–5
2.1.14 SET AUDIT Command 2–5
2.1.15 SET FILE Command 2–5
2.1.16 SET HOST/LAT Command 2–5
2.1.17 SET PROCESS Command 2–6
2.1.18 SET SECURITY Command 2–6
2.1.19 SET TERMINAL Command 2–7
2.1.20 SHOW AUDIT Command 2–7
2.1.21 SHOW DEVICES Command 2–7
2.1.22 SHOW ENTRY Command 2–7
2.1.23 SHOW ERROR Command 2–7
2.1.24 SHOW LICENSE Command 2–7
2.1.25 SHOW LOGICAL Command 2–7
2.1.26 SHOW MEMORY Command 2–7
2.1.27 SHOW PROCESS Command 2–7
2.1.28 SHOW QUEUE Command 2–8
2.1.29 SHOW SECURITY Command 2–8
2.1.30 SHOW SYSTEM Command 2–8
2.1.31 SHOW USERS Command 2–8
2.1.32 TYPE Command 2–8
2.1.33 Output Navigation Qualifiers 2–8
2.2 CONVERT/RECLAIM/KEY Qualifier 2–10

2.3 Date/Time Formats 2–10
2.4 Support for File Shelving 2–11
2.5 Help Message Utility (MSGHLP) Additions 2–11
2.5.1 Accessing $STATUS Values for Uninstalled Messages 2–12

3 System Management Features

3.1 Monitoring Performance History Program Available 3–1
3.2 Volume Shadowing for OpenVMS 3–1
3.3 RMS Journaling Supported on OpenVMS AXP Systems 3–1
3.4 Cross-Architecture Satellite Booting in VMScluster Configurations 3–2
3.5 Common AUTOGEN Parameter Files for VMScluster Systems 3–2
3.6 Clusterwide Tape Serving 3–2
3.6.1 TMSCP Tape Server 3–3
3.6.2 Loading the Magnetic Tape Server 3–3
3.6.3 TMSCP_LOAD Parameter 3–3
3.7 Clusterwide Virtual I/O Cache 3–4
3.8 New Qualifiers for the SHOW LICENSE Command 3–4
3.9 DECamds Availability Management Software 3–4
3.10 POLYCENTER Software Installation Utility 3–5
3.11 New Menu-Driven Procedure for Software Installation and Maintenance 3–5
3.12 New /EXACT_ORDER Qualifier for the BACKUP Command 3–6
3.13 Improvements to the File System 3–6
3.14 DECdtm Transaction Log Management 3–7
3.15 Error Log Utility 3–7
3.15.1 New Error Log Utility Qualifier 3–7
3.15.2 Error Formatter and Event Notification 3–7
3.16 DECevent Event Management Utility 3–7
3.17 Multiple Queue Managers 3–8
3.18 System Security Enhancements 3–9
3.18.1 Security Auditing 3–9
3.18.1.1 Distinction Between Auditing Alarms and Audits 3–10
3.18.1.2 Auditing for Additional Security Events 3–10
3.18.1.3 DCL Support for Auditing 3–10
3.18.1.4 System Service Support for Auditing 3–10
3.18.1.5 New Audit Access Control Entry 3–11
3.18.1.6 New Mechanism for Controlling Audit Messages 3–11
3.18.2 Object Protection 3–11
3.18.2.1 New Classes of Objects 3–11
3.18.2.2 Consistent Set of Security Elements for All Object Classes 3–12
3.18.2.3 Security Profile Templates 3–12
3.18.2.4 Clusterwide Distribution of Security Profiles 3–12
3.18.2.5 DCL Interfaces for Object Protection 3–12
3.18.2.6 System Services for Object Protection 3–13
3.18.2.7 Appropriate Access Types for Each Object Class 3–13
3.18.2.8 LIB$ Routines for Access Conversion 3–13
3.18.2.9 Enhanced Device Protection 3–14
3.18.3 Protected Subsystems 3–14

4.10 LOGINOUT Callout Routines 4–13
4.11 Linker Enhancement 4–13
4.11.1 New /DSF Qualifier 4–13
4.11.2 New /ATTRIBUTES Qualifier for COLLECT= Option 4–13
4.12 System Dump Analyzer for AXP 4–14
4.12.1 Using SDA CLUE Commands to Obtain and Analyze Crash Dump Information 4–14
4.12.1.1 Understanding SDA CLUE 4–14
4.12.1.2 Displaying Data Using SDA CLUE Commands 4–15
4.12.2 New Qualifier for SET PROCESS and SHOW PROCESS Commands 4–15
4.13 Digital Portable Mathematics Library (DPML) 4–15
4.14 X_Floating Atomic Data Types for Calling Routines 4–16
4.14.1 X_Floating Data Type and Datum 4–16
4.14.2 X_Floating Complex Data Type and Structure 4–17
4.15 C++ Data Type Declarations 4–18
4.15.1 C and C++ Implementations 4–18
4.16 New Run-Time Library Support Available 4–21
4.17 Native MACRO-32 Compiler 4–21
4.18 System Services Features 4–22
4.18.1 Batch and Print Queuing System Services 4–22
4.18.1.1 New Features for $GETQUI 4–22
4.18.1.2 New Features for $SNDJBC 4–22
4.18.2 Cluster Configuration Event Notification System Services 4–23
4.18.3 DECdtm System Services Enhancements 4–23
4.18.4 $FILESCAN System Service Enhancements 4–23
4.18.5 $GET_ARITH_EXCEPTION System Service 4–24
4.18.6 $GETDVI System Service Enhancements 4–24
4.18.7 New Argument for $GETUAI and $SETUAI System Services 4–25
4.18.8 $IEEE_SET_FP_CONTROL System Service 4–25
4.18.9 $SCHED System Service 4–25
4.18.10 Security System Services 4–25
4.18.11 Time Conversion Service 4–26
4.19 Support for the Movefile Subfunction 4–27
4.20 New File Information Block (FIB) Changes 4–27
4.20.1 FIB$L_ACCTL (Access Control) Subfunctions 4–27
4.20.2 IO$_CREATE (Create File) Major Functions 4–27
4.20.3 IO$_MODIFY (Modify File) Major Functions 4–28
4.21 LAT Symbiont Customization 4–29
4.22 LAT Programming Features 4–30
4.22.1 LAT $QIO Function Modifier 4–30
4.22.2 LAT SENSEMODE Argument 4–31
4.22.3 LAT SENSEMODE Item Codes 4–31
4.22.3.1 LAT SENSEMODE Item Codes for Port Node Counters Item Codes 4–31
4.22.3.2 LAT SENSEMODE Item Code for Node Counters 4–31
4.22.3.3 LAT SENSEMODE Item Codes for Protocol Errors 4–31
4.22.3.4 LAT SENSEMODE Item Code for the Service Counters Subblock 4–32
4.22.3.5 LAT SENSEMODE Item Codes for Port Entities 4–32
4.22.3.6 LAT SENSEMODE Item Codes for Queue Entries 4–32
4.22.4 LAT SETMODE Argument 4–33

3.18.4 Additional Security Enhancements . . . 3–14
3.18.4.1 Facility-Specific Rights Identifiers . . . 3–15
3.18.4.2 Rights Identifier Attributes . . . 3–15
3.18.4.3 Greater Control of Certain File Profiles . . . 3–15
3.18.4.4 New Privileges . . . 3–16
3.18.4.5 Setting Context Within SYSUAF.DAT . . . 3–16
3.18.5 Security Management Changes . . . 3–16
3.19 System Parameters . . . 3–16
3.19.1 New System Parameters for OpenVMS AXP . . . 3–16
3.19.2 New System Parameters Previously Supported by OpenVMS VAX . . . 3–18
3.20 New SYSMAN Commands: IO SET EXCLUDE and IO SHOW EXCLUDE . . . 3–20
3.20.1 IO SET EXCLUDE . . . 3–20
3.20.2 IO SHOW EXCLUDE . . . 3–20
3.21 LAT New Features . . . 3–21
3.21.1 New Features of SET HOST /LAT Command . . . 3–21
3.21.1.1 Intrasystem LAT Connections . . . 3–21
3.21.1.2 Propagation of Dialup Bit . . . 3–21
3.21.1.3 Using the SET HOST /LAT in a Command Procedure . . . 3–22
3.21.1.4 New Qualifiers to SET HOST /LAT Command . . . 3–22
3.21.2 New Features of the LAT Control Program (LATCP) Utility . . . 3–22
3.21.2.1 New LATCP Commands and Qualifiers . . . 3–22
3.21.2.2 Creating Limited Services . . . 3–23
3.21.2.3 Queuing Incoming Requests . . . 3–23
3.21.2.4 Rating Modifications . . . 3–24
3.21.2.5 Controlling Logical Name Translations . . . 3–24
3.21.2.6 Specifying Wildcards . . . 3–25

4 Programming Features

4.1 Device Support on OpenVMS AXP Systems . . . 4–1
4.2 OpenVMS AXP System-Code Debugger . . . 4–1
4.3 OpenVMS Debugger New Features . . . 4–2
4.3.1 Language Support . . . 4–3
4.3.2 Floating-Point Arithmetic Enhancements . . . 4–3
4.3.3 DECwindows Motif User Interface Enhancements . . . 4–3
4.3.4 Support for Debugging Resident Images . . . 4–4
4.3.5 Static Watchpoint Support . . . 4–5
4.3.6 DCL DEBUG Command Enhancements . . . 4–5
4.3.7 CONNECT/DISCONNECT Command Support . . . 4–6
4.3.8 OpenVMS AXP System-Code Debugger Support . . . 4–6
4.3.9 Online Help Changes . . . 4–7
4.4 OpenVMS RMS Reverse-Search Key Option . . . 4–7
4.5 Convert/Reclaim Utility Qualifiers . . . 4–8
4.5.1 New /KEY Qualifier . . . 4–8
4.5.2 Enhancement of Display from CONVERT/RECLAIM/STATISTICS Command . . . 4–9
4.6 CONV$RECLAIM Key Selection Parameter . . . 4–9
4.7 Symbolic Offset Access into Options and Statistics Arrays . . . 4–9
4.7.1 Access to Qualifier Options Array . . . 4–9
4.7.2 Access to Conversion Statistics Array . . . 4–10
4.7.3 Access to Reclamation Statistics Array . . . 4–11
4.8 New Optional Argument Supported for Calls to FDL$CREATE . . . 4–11
4.9 Programming Support for File Shelving . . . 4–12

4.22.5 LAT SETMODE Item Codes 4–33
4.22.5.1 LAT SETMODE Item Codes for Node Entities 4–33
4.22.5.2 LAT SETMODE Item Code for Service Type 4–35
4.22.5.3 LAT SETMODE Item Code for Port Type 4–35
4.23 $SETIME and DECdts 4–36
4.24 SS$_PAGRDERR Condition 4–36
4.25 LAN Programming Features 4–36
4.25.1 LAN Devices 4–36
4.25.2 Token Ring Characteristics 4–37
4.25.3 LAN Token Ring Functional Address Mapping 4–37
4.25.4 LAN Frame Formats 4–38
4.25.4.1 LAN Token Ring Frames 4–39
4.25.4.2 LAN Token Ring Source Routing Header 4–40
4.25.5 LAN Format Parameter for Token Ring 4–40
4.25.6 LAN Protocol Type PID Sharing 4–42
4.25.7 LAN Device Information 4–43
4.25.8 LAN Read with Token Ring 4–43
4.25.9 LAN Write with Token Ring 4–44
4.25.10 LAN Set Mode and Set Characteristics for Token Ring 4–46
4.25.10.1 LAN Set Controller Mode 4–46
4.25.10.2 LAN Set Mode Parameters for Packet Formats for Token Ring . . . 4–48
4.25.10.3 IO$M_SET_MAC Functional Modifier to IO$M_SETMODE 4–48
4.25.10.4 IO$M_UPDATE_MAP Functional Modifier to IO$_SETMODE with Token Ring 4–49
4.25.10.5 IO$M_ROUTE Functional Modifier to IO$_SETMODE with Token Ring 4–50
4.25.11 Sense Mode and Sense Characteristics for Token Ring 4–52
4.25.11.1 IO$M_SENSE_MAC Functional Modifier to IO$_SENSEMODE 4–52
4.25.11.2 IO$M_SHOW_MAP Functional Modifier to IO$_SENSMODE with Token Ring 4–53
4.25.11.3 IO$M_SHOW_ROUTE Functional Modifier to IO$_SENSEMODE with Token Ring 4–53

5 DECwindows Motif Display Server New Features

5.1 DECwindows Server Customization Parameters 5–1
5.1.1 Server Extensions 5–4
5.2 Shared Memory Extension Support 5–5
5.3 Using LAT Transport 5–6
5.4 AccessX Support 5–6
5.5 New Keymap Name Support 5–7
5.5.1 Keymaps of LK Version Keyboard 5–8
5.5.2 Keymaps of PC Version Keyboard 5–8
5.5.3 Simulating LK401 Function Keys 5–9
5.6 New Qualifier for the Font Compiler 5–9
5.7 Font Server Support 5–9
5.8 New Font Support 5–10

6 System Management Examples

6.1 Job Process Information (JPI) Utility 6–2
6.1.1 Installing JPI 6–4
6.1.2 Using JPI 6–6
6.2 Terminating Processes 6–14
6.3 Submitting Periodic Batch Jobs 6–15
6.4 Displaying System Roots 6–16
6.5 Deleting Directory Trees 6–18

A New OpenVMS System Messages

Index

Examples

6–1 JPI Display 6–3
6–2 Sample VMSINSTAL Log File for JPI Utility 6–5
6–3 Sample FINDROOTS Display 6–17
6–4 Sample DELTREE Display 6–19

Figures

4–1 Structure of an X_Floating Datum 4–17
4–2 Structure of an X_Floating Complex_Number Datum 4–18
4–3 LAN Frame Formats 4–39
4–4 Token Ring Frame Format 4–39
4–5 Source Routing Field 4–40
4–6 Frames with Ethernet Format 4–40
4–7 Frames with 802 Format 4–41
4–8 Frames with 802E Format 4–42
4–9 Format of the DSAP and CTL Buffer 4–45
4–10 Write Function P5 Buffer 4–45
4–11 Format of IO$M_UPDATE_MAP Setmode P2 Buffer (AXP Only) 4–49
4–12 Format of the IO$M_ROUTE P2 Buffer (AXP Only) 4–51
4–13 Format of IO$M_SHOW_MAP P2 Buffer 4–53
4–14 Format of IO$M_SHOW_ROUTE P2 Buffer 4–54

Tables

1–1 Summary of OpenVMS AXP Version 6.1 Software Features 1–1
2–1 New Keywords for the /SELECT Qualifier 2–3
2–2 F$FILE_ATTRIBUTES Items 2–4
2–3 OpenVMS System Process Privileges 2–6
2–4 Commands Affected by the Output Navigational Qualifiers 2–8
3–1 TMSCP_LOAD Parameter Settings 3–3
3–2 Access Types 3–13
3–3 New System Parameters for OpenVMS AXP Version 6.1 3–17

3–4	System Parameters Previously Supported on OpenVMS VAX	3–18
3–5	New LATCP Commands	3–22
3–6	New LATCP Command Qualifiers	3–23
4–1	IEEE Extended Floating-Point Data Type Qualifiers and Parameters	4–3
4–2	Keyed Search Combinations	4–8
4–3	CONVERT Options Array	4–9
4–4	Conversion Statistics Array	4–10
4–5	Bucket Reclamation Statistics Array	4–11
4–6	New DPML Routines	4–15
4–7	X_Floating Atomic Data Types	4–16
4–8	C and C++ Implementations	4–19
4–9	New Cluster Configuration Event Notification Services	4–23
4–10	New $FILESCAN Flag Bits	4–24
4–11	New $FILESCAN Item Codes	4–24
4–12	New $GETDVI Item Codes	4–24
4–13	Summary of New Features for Security System Services	4–26
4–14	New Timer System Services	4–26
4–15	FIB Access Control Fields	4–27
4–16	IO$_CREATE and the File Information Block	4–28
4–17	IO$_MODIFY and the File Information Block	4–28
4–18	Item Codes for Subblock LAT$_ITM_COUNTERS	4–31
4–19	LAT SENSEMODE Item Code for Node Counters	4–31
4–20	LAT SENSEMODE Item Codes for Protocol Errors	4–32
4–21	LAT SENSEMODE Item Code for the Service Counters Subblock	4–32
4–22	LAT SENSEMODE Item Codes for Port Entities	4–32
4–23	LAT SENSMODE Item Codes for Queue Entries	4–33
4–24	LAT SETMODE Item Codes for Node Entities	4–33
4–25	LAT$C_ENT_SERVICE Item Codes	4–35
4–26	LAT$C_ENT_PORT Item Codes	4–35
4–27	SS$_PAGRDERR Condition	4–36
4–28	New LAN Devices, Drivers, and Busses	4–37
4–29	Address Mappings of Token Ring Drivers	4–38
4–30	Ethernet Controller Device Characteristics	4–43
4–31	Maximum User Data Sizes	4–44
4–32	Maximum Message Sizes	4–46
4–33	P2 Attributes	4–47
4–34	Set Mode Parameters for Packet Formats	4–48
4–35	Medium Specific Parameters of IO$M_SET_MAC	4–48
4–36	Parameter ID's for P2 Buffer	4–52
4–37	Parameters of IO$M_SENSE_MAC	4–52
4–38	States of the Entry	4–55
5–1	DECwindows Customization Parameters	5–2
5–2	Loadable Server Extensions	5–5
5–3	Built-in Server Extensions	5–5
5–4	New Keymap Names	5–7
5–5	Key Mapping for LK Model Keyboards	5–8

5–6	Lucida Bright Font Names	5–10
5–7	Lucida Sans Font Names	5–12
5–8	Lucida Typewriter Font Names	5–14
5–9	MIT Miscellaneous Font Names	5–16
5–10	Type 1 Scalable	5–16
5–11	VT330 Font Names	5–16
6–1	JPI Selection Criteria	6–7
6–2	JPI Input/Output Options	6–12

Preface

Intended Audience

This manual is intended for general users, system managers, and programmers who use the OpenVMS AXP Version 6.1 operating system.

This document contains brief descriptions of the new features for OpenVMS AXP Version 6.1. For detailed information about each new feature, refer to the identified manual in the OpenVMS documentation set. For information about how some of the new features might impact your system, read the *OpenVMS AXP Version 6.1 Release Notes* before you install, upgrade, or use OpenVMS AXP Version 6.1.

Document Structure

This manual is organized as follows:

- Chapter 1 contains a summary of the new OpenVMS AXP Version 6.1 software features.
- Chapter 2 describes new features of interest to general users of the OpenVMS AXP Version 6.1 operating system.
- Chapter 3 describes new features that are applicable to the tasks performed by system managers.
- Chapter 4 describes new features that support programming tasks.
- Chapter 5 describes new features of the DECwindows Motif display server. The server is included with OpenVMS AXP Version 6.1.
- Chapter 6 presents examples of software tools useful in performing a number of common system management tasks. Copies of the tools are available on the OpenVMS AXP Version 6.1 CD–ROM.
- System messages are listed in Appendix A.

Associated Documents

Refer to the following documents for detailed information about the OpenVMS AXP Version 6.1 software features described in this manual. For more information about how to order these documents, see the *Overview of OpenVMS Documentation* or contact your Digital representative.

- *Alpha Architecture Manual*
- *Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver*
- *Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver*
- *DECamds User's Guide*

- *DECwindows Motif Version 1.2 for OpenVMS Release Notes*
- *Digital Portable Mathematics Library*
- *Guide to OpenVMS AXP Performance Management*
- *Guide to OpenVMS File Applications*
- *Managing DECwindows Motif for OpenVMS Systems*
- *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code*
- *MPH Installation Guide*
- *OpenVMS AXP CD–ROM User's Guide*
- *OpenVMS AXP Device Support: Developer's Guide*
- *OpenVMS AXP Device Support: Reference*
- *OpenVMS AXP System Dump Analyzer Utility Manual*
- *OpenVMS AXP Version 6.1 Release Notes*
- *OpenVMS AXP Version 6.1 Upgrade and Installation Manual*
- *OpenVMS Calling Standard*
- *OpenVMS DCL Dictionary*
- *OpenVMS Debugger Manual*
- *OpenVMS Guide to System Security*
- *OpenVMS I/O User's Reference Manual*
- *OpenVMS Linker Utility Manual*
- *OpenVMS Programming Concepts Manual*
- *OpenVMS Programming Interfaces: Calling a System Routine*
- *OpenVMS Record Management Services Reference Manual*
- *OpenVMS RTL Library (LIB$) Manual*
- *OpenVMS System Management Utilities Reference Manual*
- *OpenVMS System Manager's Manual*
- *OpenVMS System Messages: Companion Guide for Help Message Users*
- *OpenVMS System Services Reference Manual*
- *OpenVMS Utility Routines Manual*
- *POLYCENTER Software Installation Utility Developer's Guide*
- *POLYCENTER Software Installation Utility User's Guide*
- *RMS Journaling for OpenVMS Manual*
- *VMScluster Systems for OpenVMS*
- *VMS DECwindows Guide to Xlib Programming: MIT C Binding*
- *Volume Shadowing for OpenVMS*

Conventions

In this manual, every use of OpenVMS AXP means the OpenVMS AXP operating system, every use of OpenVMS VAX means the OpenVMS VAX operating system, and every use of OpenVMS means both the OpenVMS AXP operating system and the OpenVMS VAX operating system.

In this manual, every use of DECwindows and DECwindows Motif refers to DECwindows Motif for OpenVMS software.

The following conventions are also used in this manual:

Ctrl/*x*	A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 *x*	A sequence such as PF1 *x* indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
GOLD *x*	A sequence such as GOLD *x* indicates that you must first press and release the key defined as GOLD and then press and release another key. GOLD key sequences can also have a slash (/), dash (–), or underscore (_) as a delimiter in EVE commands. The GOLD key definition is often mapped to the PF1 key on the keypad.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
. . .	Horizontal ellipsis points in examples indicate one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	Vertical ellipsis points indicate the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.
[ ]	In command format descriptions, brackets indicate optional elements. You can choose one, none, or all of the options. (Brackets are not optional, however, in the syntax of a directory name in an OpenVMS file specification or in the syntax of a substring specification in an assignment statement.)
{ }	In command format descriptions, braces surround a required choice of options; you must choose one of the options listed.
boldface text	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason (user action that triggers a callback). Boldface text is also used to show user input in Bookreader versions of the manual.

italic text	Italic text emphasizes important information and indicates complete titles of manuals and variables. Variables include information that varies in system messages (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *device-name* contains up to five alphanumeric characters).
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`struct`	Monospace type in text identifies the following C programming language elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.
-	A hyphen in code examples indicates that additional arguments to the request are provided on the line that follows.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

1

Overview of OpenVMS AXP Version 6.1 New Features

This chapter contains an overview of the new software features supported by OpenVMS AXP Version 6.1. New software features are summarized in Table 1–1.

Table 1–1 Summary of OpenVMS AXP Version 6.1 Software Features

Feature	Description
	General Features
DCL commands	DCL commands modified to be common to OpenVMS AXP and OpenVMS VAX.
	Changes to the DEBUG command to invoke the Kept Debugger.
	New DIAGNOSE command to invoke the DECevent utility.
	New PRODUCT command to invoke the POLYCENTER Software Installation utility.
	New SET SECURITY and SHOW SECURITY commands to modify and display the security profiles of protected objects.
	Changes to the following commands: DIFFERENCES, DIRECTORY, RECALL, REPLY, SET FILE, SET HOST/LAT, SET PROCESS, SET TERMINAL, SHOW MEMORY, SHOW PROCESS, TYPE, and to the lexical functions F$FILE_ ATTRIBUTES and F$GETDVI.
	Addition of new display navigational qualifiers in the following commands: DIFFERENCES, DIRECTORY, DUMP, HELP, SEARCH, SHOW AUDIT, SHOW DEVICES, SHOW ENTRY, SHOW ERROR, SHOW LICENSE, SHOW LOGICAL, SHOW MEMORY, SHOW PROCESS, SHOW QUEUE, SHOW SYSTEM, SHOW USERS, TYPE.
New qualifier for Convert/Reclaim utility	New /KEY qualifier for the CONVERT/RECLAIM command that allows users more selectivity for reclamations and statistics.
Date/time formats	Support for date/time spellings in Chinese, Korean, and Thai languages.
Support for file shelving	OpenVMS AXP support for the layered product POLYCENTER HSM for OpenVMS.
Help Message utility additions	New /WORK_FILES and /SECTION_FILE qualifiers.
	Search paths that can now specify directories.
New system messages	Providing explanations and user actions for all new messages.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS AXP Version 6.1 Software Features

Feature	Description
	System Management Features
Monitoring Performance History Program	A new program that collects reliability data from systems being used in customer environments.
Volume Shadowing for OpenVMS	Volume shadowing available for OpenVMS AXP systems and supported in VMScluster environments that contain both VAX and AXP computers. Volume shadowing licenses are now available on a per disk basis, along with the traditional capacity (per CPU) licenses.
RMS Journaling for OpenVMS	RMS Journaling available for OpenVMS AXP systems and supported in VMScluster environments that contain both VAX and AXP computers. RMS Journaling licenses are purchased separately and are traditional capacity (per CPU) licenses.
Cross-architecture satellite booting	Feature that permits VAX boot nodes to provide boot service to AXP satellites and AXP boot nodes to provide boot service to VAX satellites in a VMScluster configuration.
AGEN$ include files	For VMScluster systems, two new include files in which common system parameters can be maintained.
TMSCP tape serving	OpenVMS AXP support for the OpenVMS tape mass storage control protocol (TMSCP) server that makes locally connected tapes accessible to all nodes in a VMScluster system.
Virtual I/O cache for VMScluster systems	Maintains system reliability while significantly improving virtual read performance for single-node or VMScluster systems.
SHOW LICENSE command	Two new qualifiers: /CLUSTER, /USAGE.
DECamds availability management software	DECamds Version 6.1 now available with the OpenVMS operating system kit.
POLYCENTER Software Installation utility	Support for a new technology that simplifies the distribution and management of software.
New menu-driven procedure for software installation and maintenance	New user interface included on the OpenVMS AXP distribution compact disc, making it easier to install or upgrade the operating system (using the POLYCENTER Software Installation utility), and to perform backup and restore operations.
New qualifier for BACKUP command	New /EXACT_ORDER qualifier.
File system performance improvements	Improved concurrency that allows multiple processes to create files on a single disk in parallel.
	Multiblock bitmap reads that reduce the time spent by the file system in allocating files.
DECdtm Transaction Log Management	New Log Manager Control Program (LMCP) utility command, CLOSE LOG.
Error Log utility	Support for a new /MODEL qualifier that allows users to create error log entries for CPUs based on their model type.
	User notified when the Error Formatter (ERRFMT) encounters a fatal error.
DECevent Event Management utility	A new utility that electively reports the contents of one or more event log files, invoked by entering the DCL command DIAGNOSE.
Batch and print queuing system features	Support for multiple queue managers.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS AXP Version 6.1 Software Features

Feature	Description
	System Management Features
Security management	Enhanced system security features through new and revised system management utilities and commands, including a single DCL interface for: • Managing object protection • Increased auditing capabilities • Support for protected subsystems
New system parameters	Addition of several new system parameters for OpenVMS AXP Version 6.1.
New SYSMAN commands	New commands to permanently exclude certain devices from autoconfiguration.
New LAT features	New features for SET HOST /LAT and LATCP.
System management examples	Examples of software tools that are useful for performing common system management tasks, including: • Displaying job process information • Terminating processes • Submitting periodic batch jobs • Displaying system roots • Deleting a directory or its contents, or both Copies of the tools are available in the [DOCUMENTATION.V061.EXAMPLES] directory on the OpenVMS AXP Version 6.1 CD–ROM.
	Programming Features
OpenVMS AXP device support	Providing information about how to write a device driver for a device connected to an AXP processor and how to load the driver into the OpenVMS AXP operating system.
OpenVMS AXP System-Code Debugger	A new programming tool to debug nonpageable system code and device drivers running at any IPL.
OpenVMS Debugger	Language support enhancements Floating-point arithmetic enhancements DECwindows Motif user interface enhancements Support for debugging resident images Static watchpoint support DCL DEBUG command enhancements CONNECT/DISCONNECT command support OpenVMS AXP System-Code Debugger support Online help changes

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS AXP Version 6.1 Software Features

Feature	Description
	Programming Features
OpenVMS RMS search key option	Support on OpenVMS AXP Version 6.1 and OpenVMS VAX Version 6.1 for a new search key option that reverses the search direction, when used with the existing search key options.
Convert/Reclaim utility DCL interface	New /KEY qualifier for the CONVERT/RECLAIM command that allows users more selectivity for doing bucket reclamations and obtaining statistics. CPU and I/O statistics displayed by the CONVERT/RECLAIM /STATISTICS command
CONV$RECLAIM utility routine parameter	Support for a new key selection parameter for the callable CONV$RECLAIM utility routine.
Callable Convert and Convert /Reclaim routines enhancement	Callable Convert and Convert/Reclaim routines enhanced to provide users with symbolic offset access to options and statistics arrays.
Argument added to FDL$CREATE routine call	Calls to FDL$CREATE that support the optional argument **dflt_fdl_spc**.
Programming support for file shelving	OpenVMS support for the layered product POLYCENTER HSM for OpenVMS.
LOGINOUT callout routines	User documentation supplied for LOGINOUT callout routines that enable users to merge OpenVMS LOGINOUT policies and events with site-specific login policies and events.
Linker utility	Enhancements including: • /DSF qualifier • /ATTRIBUTES qualifier for the COLLECT= option
CLUE commands for System Dump Analyzer (SDA) utility	SDA Crash Log Utility Extractor (CLUE) commands that obtain and analyze crash dump information.
Digital Portable Mathematics Library (DPML)	Support for 128-bit IEEE extended-precision numbers and many new routines.
IEEE X_floating atomic data types	OpenVMS AXP support for the IEEE X_floating atomic data types for passing arguments when calling routines in high-level language environments.
C++ data types	OpenVMS AXP support for C++ data type declarations for passing arguments when calling routines in high-level language environments.
New Run-Time Library support	OpenVMS Run-Time Library support for five additional high-level languages.
Native MACRO-32 compiler	Native MACRO-32 compiler now provided on OpenVMS AXP systems.
System services	System services modified to be common to OpenVMS AXP and OpenVMS VAX. Changes to the following services: $ABORT_TRANS, $CHECK_ACCESS, $CHKPRO, $END_TRANS, $FILESCAN, $GETDVI, $GETUAI, $GETQUI, $SETUAI, $SNDJBC, $START_TRANS

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS AXP Version 6.1 Software Features

Feature	Description
	Programming Features
	Addition of the following new system services: $ASCUTC, BINUTC, $CLRCLUEVT, $GET_ARITH_EXCEPTION, $GETUTC, $IEEE_SET_FP_CONTROL, $NUMUTC, $SCHED, $SETCLUEVT, $SETSHLV, $TIMCON, $TSTCLUEVT
Movefile support	Support for the movefile subfunction.
File Information Block (FIB)	Fields and subfields that now support both VAX and AXP systems: • FIB$L_ACCTL subfields • IO$_CREATE field and subfields • IO$_MODIFY field and subfields
LAT symbiont customization	Support for using the LAT protocol in a modified symbiont in the batch and print queuing system.
LAT $QIO function modifier	Support for a new LAT $QIO function modifier: IO$M_LT_QUE_CHG_NOTIF.
LAT SENSEMODE and SETMODE arguments	New Queue Entry arguments.
LAT SENSEMODE item codes	New item codes for node counters, port entities, service counters subblock, queue entries, and protocol errors.
LAT SETMODE item codes	New item codes for node entities, port type, and service type.
$SETIME and DECdts	Support for $SETIME system service and DECdts on OpenVMS VAX and OpenVMS AXP systems.
SS$_PAGRDERR condition	SS$_PAGRDERR argument that provides the I/O failure status that caused the page read error.
LAN support	New devices and drivers.
LAN Token Ring functional address mapping	Mechanism used to map Token Ring functional addresses to globally and locally administered multicast addresses.
LAN Token Ring formats	Token Ring LAN formats of frames, parameters, and fields.
LAN protocol type PID sharing	The 802E format that allows user's protocol identifier (PID) sharing.
LAN device information	Setting LAN device characteristics by using the $GETDVI system service.
LAN Token Ring read and write functions	Token Ring function codes for the transfer of read and write data.
New LAN Token Ring Set Mode /Set Characteristics and Sense Mode/Sense Characteristics	Token Ring attributes and functional modifiers for IO$_SETMODE and IO$_SENSEMODE.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS AXP Version 6.1 Software Features

Feature	Description
DECwindows Motif Display Server Features	
New features for the display server	DECwindows server customization parameters
	Shared memory extension support
	Using LAT transport
	AccessX support
	New keymap name support
	New qualifier for the font compiler
	Font server support
	New font support

2
General Features

This chapter highlights new features of general interest to OpenVMS AXP Version 6.1 users. For detailed information about a feature, refer to the identified manual in the OpenVMS AXP documentation set. In addition, refer to the *OpenVMS AXP Version 6.1 Release Notes* for information about possible system impact.

2.1 Common DCL Commands

OpenVMS AXP Version 6.1 provides new DCL commands and some changes to existing DCL commands and lexical functions. For more information about DCL commands and lexical functions see the *OpenVMS DCL Dictionary*.

2.1.1 Platform-Specific Attributes Have Been Removed

In OpenVMS AXP Version 1.5 and OpenVMS VAX Version 6.0, a number of DCL commands and lexical functions had some attributes that were specific to either OpenVMS AXP or OpenVMS VAX systems.

In OpenVMS AXP Version 6.1 and OpenVMS VAX Version 6.1, these attributes are no longer platform-specific. The following commands are now identical on both platforms:

ALLOCATE
ASSIGN/MERGE
ASSIGN/QUEUE
DELETE/ENTRY
DELETE/QUEUE
DELETE/QUEUE/MANAGER
DISABLE AUTOSTART
ENABLE AUTOSTART
F$GETQUI
F$IDENTIFIER
F$SETPRV
INITIALIZE/QUEUE
PRINT
SET AUDIT
SET DIRECTORY
SET FILE
SET HOST
SET PROCESS
SET QUEUE
SET RIGHTS_LIST
SET SECURITY (replaces SET ACL, SET PROTECTION,
and SET PROTECTION/DEVICE)
SET VOLUME
SHOW AUDIT
SHOW CLUSTER

SHOW INTRUSION
SHOW MEMORY (the /CACHE qualifier)
SHOW QUEUE/MANAGER
SHOW SECURITY (replaces SHOW ACL)
SPAWN
START/QUEUE
START/QUEUE/MANAGER
STOP/QUEUE
STOP/QUEUE/MANAGER/CLUSTER
STOP/QUEUE/NEXT
STOP/QUEUE/RESET
STOP/QUEUES/ON_NODE
SUBMIT
UNLOCK

2.1.2 DEBUG Command

The following information about the DEBUG command is new:

/KEEP
Invokes the Kept Debugger. The Kept Debugger includes a Run/Rerun capability that allows you to debug an image multiple times or debug a series of distinct images without exiting the Debugger.

Entering the DEBUG/KEEP command is the only way to invoke the Kept Debugger.

/RESUME (default)
Reinvokes the non-Kept Debugger after a Ctrl/Y key sequence has interrupted the execution of a program you are debugging. (The interrupted program must *not* have been linked with a /NOTRACEBACK qualifier on the LINK command.)

If you enter the DEBUG/RESUME command without a previous Ctrl/Y key sequence, no action occurs. Use the RUN program-name command to invoke the non-Kept Debugger.

The following example was added to the DEBUG command documentation in the *OpenVMS DCL Dictionary*:

Example

```
$ FORTRAN/DEBUG/NOOPTIMIZE WIDGET
$ LINK/DEBUG WIDGET
$ RUN/NODEBUG WIDGET

  NAME:
  NAME:
  NAME:
^Y
$ DEBUG

            [ Debugger Banner and Version ]

  %DEBUG-I-INITIAL, language is FORTRAN, module set to WIDGET
  DBG>
```

In this example, the FORTRAN and LINK commands both specify the /DEBUG qualifier to compile the program WIDGET.FOR with the Debugger symbol table information and to include the Debugger in the image file. The RUN command begins execution of the image WIDGET.EXE, which loops uncontrollably. The

Ctrl/Y key sequence interrupts the program, and the DEBUG command gives control to the Debugger.

2.1.3 DIAGNOSE Command

The DIAGNOSE command is a new command that invokes the DECevent utility and selectively reports the contents of one or more event log files. For a complete description of the DECevent utility, see the *OpenVMS System Management Utilities Reference Manual*.

2.1.4 DIFFERENCES Command

The following additions were made to the DIFFERENCES command:

- /IGNORE=CASE—allows a case-insensitive search for record differences
- Output navigation qualifiers (see Section 2.1.33).

2.1.5 DIRECTORY Command

The following additions were made to the DIRECTORY command:

2.1.5.1 New Keywords for /SELECT Qualifier

The /SELECT=(keyword[,...]) qualifier allows you to select files for display using the keywords in Table 2–1.

Table 2–1 New Keywords for the /SELECT Qualifier

Keyword	Description
[NO]ACL	Displays files that have an associated ACL or files that do not (by using the NOACL keyword). The /SELECT=FILE qualifier cannot be used with the /FULL qualifier.
FILE=(keyword[,...])	Displays specified portions of the file specification. The following keywords apply: [NO]NODE, [NO]DEVICE, [NO]DIRECTORY, [NO]NAME, [NO]TYPE, [NO]VERSION. The /SELECT=FILE qualifier cannot be used with the /FULL qualifier.
ONLINE NOONLINE	Displays files that are either on line or shelved.
SHELVABLE NOSHELVABLE	Displays only files that are either shelvable or not shelvable.

2.1.5.2 /SHELVED_STATE Qualifier

Specifies whether a file is shelved or on line. For information about file shelving see Section 2.4.

2.1.5.3 Output Navigation Qualifiers

The DIRECTORY command supports the output navigation qualifiers (see Section 2.1.33).

2.1.6 DUMP Command

The DUMP command supports the output navigation qualifiers (see Section 2.1.33).

2.1.7 F$FILE_ATTRIBUTES Lexical Function

Table 2–2 lists the new items you can specify with the F$FILE_ATTRIBUTES lexical function, the return type, and the information returned. For a complete list of F$FILE_ATTRIBUTES items, see the *OpenVMS DCL Dictionary: A–M*.

Table 2–2 F$FILE_ATTRIBUTES Items

Item Code	Return Type	Returns This Information
MOVE	String	Returns TRUE or FALSE; TRUE if movefile operations are enabled
SHELVED	String	Returns TRUE or FALSE; TRUE if the file is shelved
SHELVABLE	String	Returns TRUE or FALSE; TRUE if the file is shelvable

For more information on file shelving see Section 2.4. For more information on movefile subfunction, see *OpenVMS I/O User's Reference Manual*.

2.1.8 F$GETDVI Lexical Function

The F$GETDVI lexical function has added the DEVICE_TYPE_NAME item to the list of items returned. This item returns a string indicating the name of the device type.

2.1.9 HELP Command

The HELP command supports the output navigation qualifiers (see Section 2.1.33).

2.1.10 PRODUCT Command

The PRODUCT command is a new command that invokes the POLYCENTER Software Installation utility to do one of the following:

- Install one or more software products
- Remove one or more software products
- Perform other related operations

For a complete description of the POLYCENTER Software Installation utility and the use of the PRODUCT command, see the *POLYCENTER Software Installation Utility User's Guide*.

2.1.11 RECALL Command

The RECALL command now displays up to 254 previously entered commands on the screen for subsequent execution.

The recall buffer can hold up to 254 commands. The size of the recall buffer is 4K bytes, which will store approximately 80 commands of 48 characters each.

The following qualifiers were added to the RECALL command:

/INPUT=filespec
Causes each line of the input file to be added to the recall buffer. If you do not specify a file type, the default is .LIS. The file specification cannot include the asterisk (*) and the percent sign (%) wildcard characters.

If the /ERASE qualifier is also specified, the contents of the recall buffer are erased before the contents of the recall buffer are written from the specified file.

The /INPUT qualifier is incompatible with the /ALL and /PAGE qualifiers.

/OUTPUT=filespec
Specifies the name of the output file where the contents of the recall buffer are written. If you do not specify a file type, the default is .LIS. The file specification cannot include the asterisk (*) and the percent sign (%) wildcard characters.

If the /ERASE qualifier is also specified, the contents of the recall buffer are erased after the contents of the recall buffer are written to the specified file.

The /OUTPUT qualifier is incompatible with the /ALL and /PAGE qualifiers.

/PAGE
/NOPAGE (default)
Controls whether output from the RECALL command is displayed one screen at a time. You can cancel the display of the current file and continue with the next file by pressing Ctrl/Z.

The /PAGE qualifier is incompatible with the /INPUT and /OUTPUT qualifiers.

2.1.12 REPLY Command

Message pertaining to LICENSES have been added to the list of messages that can be displayed or disabled (using /ENABLE and /DISABLE) from broadcasting to terminals.

2.1.13 SEARCH Command

The SEARCH command supports the output navigation qualifiers (see Section 2.1.33).

2.1.14 SET AUDIT Command

The system can now audit ill-formed internal calls (identified by NSA$M_INTERNAL) to the $AUDIT_EVENT, $CHECK_PRIVILEGE, $CHKPRO, or $CHECK_ACCESS system services. These events indicate that some privileged code (having at least AUDIT privilege) has failed to properly audit some event. This may be a serious problem for managing system security, but does not indicate a breach in system security.

This auditing may be controlled using the AUDIT=ILLFORMED keywords with /ENABLE or /DISABLE qualifiers.

2.1.15 SET FILE Command

The SET FILE command has new /SHELVABLE and /NOSHELVABLE qualifiers that control whether a file is shelvable. For more information on file shelving see Section 2.4.

2.1.16 SET HOST/LAT Command

The SET HOST/LAT command has the following new qualifiers:

/AUTOPROMPT (default)
/NOAUTOPROMPT
This qualifier causes an OpenVMS "Username:" prompt to appear with no additional user action when a SET HOST/LAT command is issued.

On a terminal server port, you can configure the port to have AUTOPROMPT disabled (/NOAUTOPROMPT) so that you are required to press Return when connecting to a node to get the "Username:" prompt. However, when connecting to a reverse LAT service, the AUTOPROMPT characteristic should be disabled.

/QUEUE
/NOQUEUE (default)
When connecting to a reverse LAT service that is already in use (such as a dial-out modem), you are notified that the service is in use and the SET HOST/LAT command terminates. However, the LAT service can allow incoming connections to be queued to any reverse LAT service that supports service queuing, which enables users to be placed on a queue when using reverse LAT services.

The LAT service reports your position in the queue. When the resource becomes available, you are immediately notified. You can cancel the queued connection by pressing Ctrl/Y and terminating the SET HOST /LAT command.

2.1.17 SET PROCESS Command

The SET PROCESS command has the following new qualifier.

/AUTO_UNSHELVE
/NOAUTO_UNSHELVE
/AUTO_UNSHELVE controls whether the process automatically unshelves files. For more information on file shelving see Section 2.4.

The SET PROCESS command has the following modified qualifier:

/PRIVILEGES=(privilege[,...])
Requires SETPRV privilege as an authorized privilege to enable any privilege you do not have in your authorized privilege mask.

Table 2–3 lists the process privileges available on all OpenVMS platforms.

Table 2–3 OpenVMS System Process Privileges

ACNT	ALLSPOOL	ALTPRI	AUDIT
BUGCHK	BYPASS	CMEXEC	CMKRNL
DETACH	DIAGNOSE	DOWNGRADE	EXQUOTA
GROUP	GRPNAM	GRPPRV	IMPORT
LOG_IO	MOUNT	NETMBX	OPER
PFNMAP	PHY_IO	PRMCEB	PRMGBL
PRMMBX	PSWAPM	READALL	SECURITY
SETPRV	SHARE	SHMEM	SYSGBL
SYSLCK	SYSNAM	SYSPRV	TMPMBX
UPGRADE	VOLPRO	WORLD	

2.1.18 SET SECURITY Command

The SET SECURITY command modifies the access control list (ACL), protection code, and owner of an object.

2.1.19 SET TERMINAL Command

The following SET TERMINAL command qualifiers are new or modified:

/COLOR
Sets the ANSI_COLOR terminal characteristic and identifies the terminal as capable of supporting the ANSI color escape sequences.

/DEC_CRT[=(value1,value2,value3)]
/NODEC_CRT[=(value1,value2,value3)]
Controls whether the terminal conforms to Digital VT100-, VT200-, VT300-, VT400-, or VT500-family standards and supports the minimum standards, including the additional Digital escape sequences.

2.1.20 SHOW AUDIT Command

The SHOW AUDIT command supports the output navigation qualifiers (see Section 2.1.33).

2.1.21 SHOW DEVICES Command

The SHOW DEVICES command supports the output navigation qualifiers (see Section 2.1.33).

2.1.22 SHOW ENTRY Command

The SHOW ENTRY command supports the output navigation qualifiers (see Section 2.1.33).

2.1.23 SHOW ERROR Command

The SHOW ERROR command supports the output navigation qualifiers (see Section 2.1.33).

2.1.24 SHOW LICENSE Command

The SHOW LICENSE command supports the output navigation qualifiers (see Section 2.1.33).

2.1.25 SHOW LOGICAL Command

The SHOW LOGICAL command supports the output navigation qualifiers (see Section 2.1.33).

2.1.26 SHOW MEMORY Command

The SHOW MEMORY command supports the output navigation qualifiers (see Section 2.1.33).

Also, the /CACHE qualifier is now common to OpenVMS platforms.

2.1.27 SHOW PROCESS Command

The SHOW PROCESS command has a modified /INTERVAL qualifier and a new /SHELVING qualifier:

/INTERVAL=n
Use /INTERVAL=n with the /CONTINUOUS qualifier to update the information at the specified number of seconds.

/SHELVING
/SHELVING displays whether the process automatically unshelves files.

The SHOW PROCESS command also supports the output navigation qualifiers (see Section 2.1.33).

2.1.28 SHOW QUEUE Command

The SHOW QUEUE command supports the output navigation qualifiers (see Section 2.1.33).

2.1.29 SHOW SECURITY Command

The SHOW SECURITY command displays the name, class, and profile of a protected object.

2.1.30 SHOW SYSTEM Command

The SHOW SYSTEM command supports the output navigation qualifiers (see Section 2.1.33).

2.1.31 SHOW USERS Command

The SHOW USERS command supports the output navigation qualifiers (see Section 2.1.33).

2.1.32 TYPE Command

When you use the TYPE command and specify a file name and not a file type, the file type defaults to .LIS. However, you can specify different default file type with the TYPE$DEFAULT_FILESPEC logical name.

The TYPE command supports the output navigation qualifiers (see Section 2.1.33).

2.1.33 Output Navigation Qualifiers

The commands listed in Table 2–4 support the output navigation qualifiers on OpenVMS systems.

Table 2–4 Commands Affected by the Output Navigational Qualifiers

Command	/EXACT	/HEADER	/HIGHLIGHT	/PAGE	/SEARCH	/WRAP
DIFFERENCES	x		x	x	x	x
DIRECTORY	x		x	x	x	x
DUMP	x		x	x	x	x
HELP	x		x	x	x	x
SEARCH			x	x		x
SHOW AUDIT	x		x	x	x	x
SHOW DEVICES	x		x	x	x	x
SHOW ENTRY	x		x	x	x	x
SHOW ERROR	x		x	x	x	x
SHOW LICENSE	x		x	x	x	x
SHOW LOGICAL	x		x	x	x	x
SHOW MEMORY	x		x	x	x	x

(continued on next page)

Table 2–4 (Cont.) Commands Affected by the Output Navigational Qualifiers

Command	/EXACT	/HEADER	/HIGHLIGHT	/PAGE	/SEARCH	/WRAP
SHOW PROCESS	x		x	x	x	x
SHOW QUEUE	x		x	x	x	x
SHOW SYSTEM	x		x	x	x	x
SHOW USERS	x		x	x	x	x
TYPE	x	x	x	x	x	x

The following section describes the output navigation qualifiers.

/EXACT
Use with the /PAGE=SAVE and /SEARCH qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

If you specify the /EXACT qualifier without the /SEARCH qualifier, exact search mode is enabled when you set the search string with the Find (E1) key.

/HEADER
/NOHEADER
Specifies whether a header line is placed at the top of each file that is displayed with the TYPE /PAGE=SAVE command. (Use with the TYPE command only.)

/HIGHLIGHT[=keyword]
/NOHIGHLIGHT (default)
Use with the /PAGE=SAVE and /SEARCH qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

/PAGE[=keyword]
/NOPAGE (default)
Controls the display of information on the screen.

You can use the following keywords with the /PAGE qualifier:

CLEAR_SCREEN	Clears the screen before each page is displayed.
SCROLL	Displays information one line at a time.
SAVE[=*n*]	Enables screen navigation of information, where *n* is the number of pages to store.

The /PAGE=SAVE qualifier allows you to navigate through screens of information. The /PAGE=SAVE qualifier stores up to 5 screens of up to 255 columns of information. When you use the /PAGE=SAVE qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow (↑), Ctrl/B	Scroll up one line.
Down arrow (↓)	Scroll down one line.
Left arrow (←)	Scroll left one column.

Key Sequence	Description
Right arrow (→)	Scroll right one column.
Find (E1)	Specify a string to find where the information is displayed.
Insert Here (E2)	Scroll right one half screen.
Remove (E3)	Scroll left one half screen.
Select (E4)	Toggle 80/132 column mode.
Prev Screen (E5)	Get the previous page of information.
Next Screen (E6), Return, Enter, Space	Get the next page of information.
F10, Ctrl/Z	Exit. (Some utilities define these differently.)
Help (F15)	Display utility help text.
Do (F16)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.
N	Display the next file (when more than one file is specified with the TYPE command).
Q	Quit displaying output from the TYPE command.

The /PAGE qualifier is not compatible with the /OUTPUT qualifier.

/SEARCH="string"
Use with the /PAGE=SAVE qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the /SEARCH qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find (E1) key while the information is being displayed. Quotation marks are not required for a dynamic search.

/WRAP
/NOWRAP (default)
Use with the /PAGE=SAVE qualifier to limit the number of columns to the width of the screen and to wrap lines that extend beyond the width of the screen to the next line.

The /NOWRAP qualifier extends lines beyond the width of the screen and can be seen when you use the scrolling (left and right) features provided by the /PAGE=SAVE qualifier.

2.2 CONVERT/RECLAIM/KEY Qualifier

You can now specify the /KEY qualifier with the CONVERT/RECLAIM command, allowing you more selectivity in doing bucket reclamations and obtaining related statistics. See Section 4.5 for more information.

2.3 Date/Time Formats

Special date/time spellings are now supported for four new languages. Users or application programmers can select the desired language by defining the logical name, SYS$LANGUAGES. The following table lists the new languages and their equivalent names defined by SYS$LANGUAGES:

Language	Equivalent Name
Chinese (simplified character)	Hanzi
Chinese (traditional character)	Hanyu
Korean	Hangul
Thai language	Thai

For example, if you want the spelling for Korean and Thai languages to be defined, define SYS$LANGUAGES prior to invoking LIB$DT_STARTUP.COM, as shown in the following example:

```
$ DEFINE SYS$LANGUAGES HANGUL,THAI
$ @SYS$MANAGER:LIB$DT_STARTUP
```

For more detailed information on date/time formats, see the *OpenVMS AXP Version 6.1 Release Notes*.

2.4 Support for File Shelving

OpenVMS AXP Version 6.1 supports the layered product POLYCENTER HSM for OpenVMS. This is a file shelving facility that lowers the average cost of storing data that is rarely used.

File shelving allows users and processes to move an inactive file to nearline or offline storage transparently, and to return the file to online storage when needed. The following table defines these terms:

Term	Definition
Online storage	Data is stored so that it is immediately available to the user without human or robotic intervention.
Offline storage	Data is stored so that human intervention is required to make it available to the user.
Nearline storage	Data is stored so that the intervention of a robot is required to make it available to the user.

2.5 Help Message Utility (MSGHLP) Additions

The Help Message utility (MSGHLP) has three new features:

- You can now specify a directory name in a search path. If no file name is associated with a directory, Help Message searches all .MSGHLP$DATA files in that directory. This allows you to add or delete files in a specified directory without having to change the search path definition.
- You can specify /WORK_FILES=*nn* with /SORT in order to use work files on large sort operations that previously failed.
- You can specify the new /SECTION_FILE qualifier to identify message section files to the system so that Help Message can interpret $STATUS values for messages pointed to by those files. See Section 2.5.1 for details.

2.5.1 Accessing $STATUS Values for Uninstalled Messages

Any messages that are not installed as part of the OpenVMS operating system cannot be equated with a value stored in $STATUS until they are recognized by the system. Such messages include user-supplied messages, third-party messages, and messages from layered products and certain other OpenVMS facilities. Your system manager may already have linked all the messages you need at the system level. If not, you can make these links in your own account. (System managers who want information about linking message section files at the system level should refer to the *OpenVMS System Manager's Manual*.)

Before linking such messages, determine whether you must frequently access them using the value in $STATUS or a value specified using the /STATUS qualifier. If not, the simplest alternative is to use the HELP/MESSAGE command and specify the message identifier or message text in the search string.

If you prefer the convenience of having Help Message interpret a $STATUS value, there are two ways that you can make the system recognize uninstalled messages:

- The simplest method is to use the SET MESSAGE command to link an unknown messages .EXE file. For example:

  ```
  $ SET MESSAGE TOOLS:[MYPROJ]MYNEWMSG.EXE
  ```

  This command includes the messages pointed to by file MYNEWMSG.EXE as part of the system until you enter another SET MESSAGE command specifying a different file.

- For a more permanent solution, execute the following command for every message section file that you would like to include in the list of files that Help Message automatically searches for $STATUS values:

  HELP/MESSAGE/SECTION_FILE=disk:[directory]file-name.EXE

  By default, /SECTION_FILE tries to generate or update object library SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.OLB. Unless you have write access to system files, you must first define the logical name MSGHLP$MESSAGE_SECTIONS to point to your own account.

  For example, to make the messages pointed to by file TOOLS:[MYPROJ]MYNEWMSG.EXE known in your own account, use the following command sequence:

  ```
  $ DEFINE MSGHLP$MESSAGE_SECTIONS SYS$LOGIN:MSGHLP$MESSAGE_SECTIONS
  $ HELP/MESSAGE/SECTION_FILE=TOOLS:[MYPROJ]MYNEWMSG.EXE
  ```

  Unlike the SET MESSAGE command, the /SECTION_FILE qualifier permanently links the specified message section file unless you later delete it.

  The default device and directory for the file specification is SYS$MESSAGE, where all OpenVMS message section files are stored, and the default file type is .EXE. Therefore, the following command automatically includes all OpenVMS message section files:

  ```
  $ HELP/MESSAGE/SECTION_FILE=*
  ```

  Your system manager may already have included the OpenVMS message section files at the system level. You will need to add these only if you define the logical name MSGHLP$MESSAGE_SECTIONS to point somewhere other than SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.

Help Message can search a maximum of 42 message section files. Files are searched in the following order:

- The last file activated by a SET MESSAGE command (if any).
- The message section files that Help Message is linked with before it is shipped:
  - SYSMSG.EXE
  - SYSMGTMSG.EXE
  - CLIUTLMSG.EXE
  - PRGMSG.EXE
- Any message section files explicitly linked against MSGHLP$MAIN.EXE or any of the shareable images that it references.
- Any nonduplicate message section files linked with a MSGHLP$MESSAGE_SECTIONS.EXE file, if such a file exists.

  The MSGHLP$MESSAGE_SECTIONS.EXE file is automatically generated from the MSGHLP$MESSAGE_SECTIONS.OLB file each time the HELP/MESSAGE command is executed with the /SECTION_FILE qualifier. Each execution of the /SECTION_FILE qualifier adds a module to the .OLB file; each module represents a message section file.

  Message section files are searched in the order in which they are listed in the .OLB file. (Order is alphabetical.) If the total count of message section files exceeds 42, message section files listed near the end of the .OLB file might not be searched.

  To display the contents of your .OLB file, use the following command:

```
$ LIBRARY/LIST MSGHLP$MESSAGE_SECTIONS.OLB
```

  The names of the modules in this .OLB file are derived from strings specified in the /SECTION_FILE qualifier.

  You can remove references to message section files in the .OLB file in order to control which 42 message section files Help Message will search. For example, to delete the NCP (Network Control Program) messages from your private .OLB file, use the following command:

```
$ LIBRARY/DELETE=NETWRKMSG MSGHLP$MESSAGE_SECTIONS.OLB
```

  If you delete a module from the .OLB file, you must execute the HELP/MESSAGE command with the /SECTION_FILE qualifier in order to generate an updated .EXE file. The qualifier argument can specify either a new file or a file that is already listed in the .OLB file.

For complete information about the Help Message utility, refer to *OpenVMS System Messages: Companion Guide for Help Message Users*.

3
System Management Features

This chapter highlights new system management features for OpenVMS AXP Version 6.1. For detailed information about a feature, refer to the identified manuals in the OpenVMS AXP documentation set. In addition, refer to the *OpenVMS AXP Version 6.1 Release Notes* for information about possible system impact.

Examples of software tools that are useful for performing some common system management tasks are described in Chapter 6.

3.1 Monitoring Performance History Program Available

Digital is committed to providing highly reliable systems. OpenVMS AXP Version 6.1 includes the Monitoring Performance History (MPH) program that collects reliability data from systems being used in customer environments. The program periodically collects data and sends it to Digital for analysis.

If you are interested in participating in this program, refer to the *OpenVMS AXP Version 6.1 Release Notes* for more information about MPH and how to install it. The *MPH Installation Guide* is available with the MPH kit.

3.2 Volume Shadowing for OpenVMS

In addition to running on OpenVMS VAX systems, volume shadowing is now available on OpenVMS AXP systems. This means that you can implement volume shadowing in a VMScluster environment that contains both VAX and AXP systems.

Volume shadowing licenses are now available on a per-disk basis, along with the traditional capacity (per CPU) licenses.

For more detailed information, see *Volume Shadowing for OpenVMS*.

3.3 RMS Journaling Supported on OpenVMS AXP Systems

In addition to running on OpenVMS VAX systems, RMS Journaling is now available on OpenVMS AXP systems. This means that you can implement RMS Journaling in a VMScluster environment that contains both VAX and AXP systems. After-image, before-image, and recovery unit journaling are all supported on both platforms.

RMS Journaling licenses are sold separately from the operating system license and are available on a traditional capacity (per CPU) basis.

For more detailed information, see *RMS Journaling for OpenVMS Manual*.

3.4 Cross-Architecture Satellite Booting in VMScluster Configurations

Cross-architecture satellite booting permits VAX boot nodes to provide boot service to AXP satellites and AXP boot nodes to provide boot service to VAX satellites. For some VMScluster configurations, cross-architecture boot support can simplify day-to-day system operations and reduce the complexity of managing VMScluster environments that include both VAX and AXP systems.

Cross-architecture booting and VMScluster systems are described in *VMScluster Systems for OpenVMS*.

3.5 Common AUTOGEN Parameter Files for VMScluster Systems

When adding a node or a satellite to a VMScluster system, the CLUSTER_CONFIG.COM procedure adds one of the following lines in the MODPARAMS.DAT file:

IF the node being added is a...	THEN...
Satellite node	CLUSTER_CONFIG.COM adds the following line to the MODPARAMS.DAT file:
	`AGEN$INCLUDE_PARAMS SYS$MANAGER:AGEN$NEW_SATELLITE_DEFAULTS.DAT`
Nonsatellite node	CLUSTER_CONFIG.COM adds the following line to the MODPARAMS.DAT file:
	`AGEN$INCLUDE_PARAMS SYS$MANAGER:AGEN$NEW_NODE_DEFAULTS.DAT`

The AGEN$NEW_SATELLITE_DEFAULTS.DAT and AGEN$NEW_NODE_DEFAULTS.DAT files hold AUTOGEN parameter settings that are common to all satellite nodes or nonsatellite nodes in the cluster. Use of these files simplifies system management, because you can maintain common system parameters in either one or both of these files. When you are adding or changing the common parameters, this capability eliminates the need to make modifications in the MODPARAMS.DAT files located on every node in the cluster.

Initially, these files contain no parameter settings. You edit the AGEN$NEW_SATELLITE_DEFAULTS.DAT and AGEN$NEW_NODE_DEFAULTS.DAT files, as appropriate, to add, modify, or edit system parameters. For example, you might edit the AGEN$NEW_SATELLITE_DEFAULTS.DAT file to set the MIN_GBLPAGECNT parameter to 5000. AUTOGEN makes the MIN_GBLPAGECNT parameter and all other parameter settings in the AGEN$NEW_SATELLITE_DEFAULTS.DAT file common to all satellite nodes in the cluster.

AUTOGEN uses the parameter settings in the AGEN$NEW_SATELLITE_DEFAULTS.DAT or AGEN$NEW_NODE_DEFAULTS.DAT files the first time AUTOGEN is run and with every subsequent execution.

See *VMScluster Systems for OpenVMS* for complete information about these include files and about setting system parameters for VMScluster systems.

3.6 Clusterwide Tape Serving

Clusterwide tape serving is now supported by OpenVMS AXP systems.

3.6.1 TMSCP Tape Server

Included in OpenVMS AXP Version 6.1 is the OpenVMS tape mass *storage* control protocol (TMSCP) server. The tape server allows the system manager to make locally connected tape drives accessible clusterwide. A cluster-accessible tape is a tape that every node in the cluster can recognize and access.

The tape server allows nodes without a locally connected tape drive to gain direct access to a tape drive connected to another node.

Once the server has been loaded and tape devices have been set as served, the devices can be accessed from any node in the VMScluster environment using DCL commands. INITIALIZE, MOUNT, and BACKUP operations can be done on remote tape devices in the same way as they are currently done using locally connected devices.

Note

Tape drives are not shared devices. Only one user can access a tape at a time. With the tape server, served tape drives are accessible to all nodes in a cluster but can be allocated and accessed by only one process at a time.

See *VMScluster Systems for OpenVMS* for details about implementing the TMSCP server.

3.6.2 Loading the Magnetic Tape Server

By default, OpenVMS does not load the tape server software. To use the server, the system manager must modify the system parameter TMSCP_LOAD and, optionally, the TAPE_ALLOCLASS parameter.

3.6.3 TMSCP_LOAD Parameter

The TMSCP_LOAD system parameter loads the TMSCP server software and sets up any locally connected tapes to be served.

Table 3–1 describes the two possible settings for the TMSCP_LOAD parameters.

Table 3–1 TMSCP_LOAD Parameter Settings

Setting	Result
0	Do not load the TMSCP tape server. Do not serve any local tape devices clusterwide.
1	Load the TMSCP tape server at the time the system is booted. Serve all local TMSCP tape devices clusterwide.

The TMSCP system parameter serves TMSCP tape drives only, including:

- DSSI tapes such as the TF85
- Tape devices connected to an HSC
- Tape devices connected to an HSJ

You can use the SHOW DEVICE command to identify TMSCP tape devices. Use the SHOW DEVICE M command to obtain a list of tape (and mailbox) devices. MU and MI tape devices are TMSCP tapes, so if SHOW DEVICE M displays a TU83 tape device as MUA0, the device is a TMSCP device.

You can also use the OpenVMS System Dump Analyzer (SDA) utility to determine if a tape device is a TMSCP tape. To invoke SDA, enter ANALYZE/SYSTEM from a privileged account at the DCL prompt. Then, use the following SDA command:

SHOW DEVICE MU*cn*

where:

c is the controller letter
n is the device unit number

3.7 Clusterwide Virtual I/O Cache

The virtual I/O cache is a clusterwide, file-oriented disk cache that reduces I/O bottlenecks and improves performance. The virtual I/O cache operates transparently to system management and application software. This new feature maintains system reliability while it significantly improves virtual read performance.

The virtual I/O cache works on all supported configurations, from single-node systems to large mixed-interconnect VMScluster systems. Any clustered or standalone system can take advantage of the virtual I/O cache; no additional license is required. For more information about the virtual I/O cache, see the *Guide to OpenVMS AXP Performance Management*.

3.8 New Qualifiers for the SHOW LICENSE Command

Two new qualifiers are available for the SHOW LICENSE command. The /USAGE qualifier displays information on the current license use for a specific product installed on the system. It tells you how many license units are loaded, how many are currently allocated, and how many are currently available, as well as the license type.

The /CLUSTER qualifier, in conjunction with the /UNIT_REQUIREMENTS qualifier, displays the license unit requirements for every node in a VMScluster system.

For more detailed information, see the *OpenVMS DCL Dictionary*.

3.9 DECamds Availability Management Software

The DECamds Version 6.1 kit is now included with the base OpenVMS operating system kit as a separately installable kit. DECamds is a real-time monitoring, investigation, diagnostic, and software correction tool to assist system managers and system integrators in improving system availability on OpenVMS systems. DECamds collects and analyzes system-level data simultaneously from multiple nodes, directing all output to a centralized DECwindows Motif for OpenVMS display. On the basis of collected data, DECamds analyzes and directs corrective actions for exhausted resources and problems with gaining access to system resources.

DECamds is comprised of two components: the DECamds console application and the DECamds data collecting driver. For OpenVMS AXP Version 6.1, only the DECamds driver support is provided. Therefore, if you have only OpenVMS AXP systems, do not install this product. The DECamds console can be installed on an OpenVMS VAX system. Refer to the *OpenVMS AXP Version 6.1 Release Notes* for information on installing DECamds software kits in a mixed-architecture VMScluster environment.

For information on how to install and use DECamds software, refer to the *DECamds User's Guide*, which is included in the OpenVMS documentation kit. You must have either the VMScluster license or the DECamds Version 1.0 license to use DECamds.

3.10 POLYCENTER Software Installation Utility

The POLYCENTER Software Installation architecture is a new technology that simplifies the distribution and management of software.

For system managers, the POLYCENTER Software Installation utility provides DIGITAL Command Language (DCL) and DECwindows Motif interfaces that can be used to configure, install, and remove software products. It also allows system managers to track the status of software on their system.

For software providers, the POLYCENTER Software Installation utility simplifies the task of packaging software for the following reasons:

- The utility provides a standard process and toolset for packaging products on all platforms.
- All software and documentation can exist on single pieces of media.
- The utility keeps track of which products (and versions) have been installed and removed in the execution environment. By using this information, you can design your installation procedure so that users can use the utility to manage version dependencies.
- Installations require less code than most conventional installation methods. This results in performance gains and reduced development time over conventional installations. The code is also nonprocedural, which means that you do not need to understand the details of how the utility functions.
- You can include installation text to guide users through an installation, resulting in a higher installation success rate.

For more detailed information, see the *POLYCENTER Software Installation Utility User's Guide* and the *POLYCENTER Software Installation Utility Developer's Guide*.

3.11 New Menu-Driven Procedure for Software Installation and Maintenance

The OpenVMS AXP distribution compact disc includes a menu-driven command procedure that allows you to easily upgrade or install the operating system and to perform related operations such as backing up the system disk. This command procedure starts automatically when you boot the OpenVMS AXP distribution compact disc, displaying a menu from which you choose options to perform the following tasks:

- Install or upgrade the operating system, using the POLYCENTER Software Installation utility
- Enter a DCL environment, from which you can perform preinstallation or maintenance tasks such as mounting or showing devices and backing up or restoring files on the system disk

For more detailed information about using the new menu-driven procedure, see the *OpenVMS AXP Version 6.1 Upgrade and Installation Manual*. For information about using the menu to perform backup and restore operations, see the *OpenVMS System Manager's Manual*.

3.12 New /EXACT_ORDER Qualifier for the BACKUP Command

For OpenVMS AXP Version 6.1, BACKUP has a new output qualifier, /EXACT_ORDER. Depending on the other qualifiers you specify on the command line, the /EXACT_ORDER qualifier enables you to do the following:

- Specify the exact order of volume labels that you want to use in a BACKUP operation.
- Preserve the existing volume label on a tape.
- Prevent previous volumes of a multivolume save operation from being overwritten.

Note the following restrictions when you use the /EXACT_ORDER output qualifier:

- If you use the /EXACT_ORDER qualifier, you cannot specify a label longer than 6 characters on the command line. If you specify a label longer than 6 characters, BACKUP displays the following error message:

  ```
  %BACKUP-F-INVQUAVAL, value 'NET_CONTROL' invalid for /LABEL qualifier
  ```

- You cannot use the /IGNORE=LABEL_PROCESSING qualifier with the /EXACT_ORDER qualifier.

For more information, see the *OpenVMS System Management Utilities Reference Manual: A–L*.

3.13 Improvements to the File System

OpenVMS AXP Version 6.1 includes an improved file system. This improved file system is also available as part of OpenVMS VAX Version 6.1, and as a special release, known as the XQP+ for PATHWORKS.

The improved file system provides the following new features to improve the performance of OpenVMS:

New Feature	Description
Improved concurrency	Allows multiple processes to create files on a single disk in parallel. Previously, only one process could create files on a disk at a time.
Multiblock bitmap reads	Reduces the time spent by the file system in allocating files.

You can control improved concurrency using the static system parameter, XQPCTL2. Improved concurrency is off by default. To turn it on, set XQPCTL2 to 1.

Caution

If you are in a VMScluster environment, do not turn on improved concurrency unless all the nodes are running the enhanced file system. Each node in the VMScluster environment must be running one of the following:

- OpenVMS AXP Version 6.1
- OpenVMS VAX Version 6.1
- XQP+ for PATHWORKS

3.14 DECdtm Transaction Log Management

The new Log Manager Control Program (LMCP) utility command, CLOSE LOG, lets you perform the following tasks without rebooting any nodes:

- Change the size of a transaction log
- Move a transaction log
- Dismount a disk that has transaction logs
- Disable DECdtm services

For more detailed information, see the *OpenVMS System Manager's Manual.*

3.15 Error Log Utility

New features supported by the Error Log utility include the /MODEL qualifier and user notification of fatal errors encountered by the Error Formatter.

3.15.1 New Error Log Utility Qualifier

The qualifier /MODEL, a new feature of the Error Log utility, is used on OpenVMS AXP systems only. It allows you to create error log entries for CPUs based on their model type.

For more detailed information, see the *OpenVMS System Management Utilities Reference Manual.*

3.15.2 Error Formatter and Event Notification

The Error Formatter, ERRFMT, provides a new feature that allows you to send mail to a user if the ERRFMT process encounters a fatal error and deletes itself. Sending mail is enabled by default. However, two new system logical names ERRFMT$_SEND_MAIL and ERRFMT$_SEND_TO defined in the SYS$STARTUP:VMS$CONFIG-050_ERRFMT.COM file control this feature. See the *OpenVMS System Manager's Manual* for more information.

3.16 DECevent Event Management Utility

The DECevent Event Management (DECevent) utility for the OpenVMS AXP operating system provides the interface between a system user and the system's event log files. DECevent lets users produce ASCII reports derived from system event entries. The format of the ASCII reports depends on the command entered on the command line interface with a maximum character limit of 255 characters. Event report information can be filtered by event types, date, time, and event entry numbers. Event report formats can be selected from full disclosure to very brief information messages. The include and exclude qualifiers provide a wide range of selection criteria to narrow down the focus of event searches.

DECevent also offers an interactive command shell interface that recognizes the same commands used at the command line interface. From the interactive command shell, users can customize, change, or save system settings.

DECevent uses the system event log file, SYS$ERRORLOG:ERRLOG.SYS, as the default input file for event reporting, unless another file is specified.

DECevent provides the following report types:

- Full
- Brief
- Terse
- Summary
- FSTERR

Used with qualifiers, these report types allow a user to view event information in several ways.

For more detailed information, see the *OpenVMS System Manager's Manual.*

3.17 Multiple Queue Managers

With OpenVMS AXP Version 1.5, a single queue manager managed batch and print queuing activity for an entire VMScluster system. OpenVMS AXP Version 6.1 allows you to create multiple queue managers. This feature is useful in clusters with CPU or memory shortages. It allows you to distribute the batch and print work load across different nodes in the cluster. For example, you might create separate queue managers for batch queues and print queues. You could then run the batch queue manager on one node and the print queue manager on a different node.

This feature is available with the following restrictions:

- The multiple queue manager feature cannot be used until all nodes in a VMScluster environment are running OpenVMS AXP Version 6.1 or all nodes in a dual-architecture VMScluster environment are running OpenVMS AXP Version 6.1 and OpenVMS VAX Version 6.0 or Version 6.1.
- Once you begin using multiple queue managers, you cannot bring into the VMScluster environment a node running an OpenVMS version earlier than OpenVMS AXP Version 6.1 or OpenVMS VAX Version 6.0. Doing so might cause unexpected results and failures.
- Queues running on one queue manager cannot reference queues running on a different queue manager. For example, a generic queue running on queue manager A cannot direct jobs to an execution queue running on queue manager B. In addition, you cannot move a job from a queue on queue manager A to a queue on queue manager B.
- No more than five queue managers are allowed in a VMScluster environment.

This feature, previously documented for the OpenVMS VAX system, is now also supported by OpenVMS AXP Version 6.1. For a complete description of this feature, see the *OpenVMS System Manager's Manual: Essentials.*

New DCL commands allow you to manage multiple queue managers. For more information, see Section 2.1 in this manual and the *OpenVMS DCL Dictionary.*

3.18 System Security Enhancements

This section highlights new system security features of interest to OpenVMS AXP Version 6.1 users. Except for the DECnet subsystem, OpenVMS AXP Version 6.1 contains all the security features of OpenVMS VAX 6.0, which was successfully evaluated by the United States government for C2 compliance.

The following enhancements are in the areas of auditing and object protection:

- The system's auditing capabilities have increased. You can audit many more classes of security-relevant events, you can select an event as either an alarm or an audit, you can add Audit access control entries (ACEs) to objects, and you can control the flow of audit messages using a new SET AUDIT qualifier (/BACKLOG).
- The system now protects more object classes. New classes of protected objects include volume, common event flag cluster, resource domain, and security class. All protected objects now have a consistent set of security elements: an access control list (ACL), an owner, and a protection code. In addition, all object classes support full ACL auditing.
- You can modify or display the security profile of all objects by using one set of DCL commands, SET SECURITY and SHOW SECURITY.
- Security administrators can now control access to files and other objects by using protected subsystems. Protected subsystems provide an alternative to creating installed privileged images or protected shareable images (user-written system services).

Along with these security changes, the following enhancements are provided:

- Expanded use of rights identifiers (see Section 3.18.4.1 and Section 3.18.4.2)
- Greater control of files created within directories owned by a resource identifier (see Section 3.18.4.3)
- Added privileges to support auditing and SEVMS, a product that enhances the security of OpenVMS systems (see Section 3.18.4.4)

For detailed information about any of the changes described in the following sections, see the *OpenVMS Guide to System Security*. In addition, refer to the *OpenVMS AXP Version 6.1 Release Notes* for information about possible system impact.

3.18.1 Security Auditing

OpenVMS AXP Version 6.1 includes changes to security auditing, which are detailed in this section. They include:

- Independent reporting of events as alarms or audits
- More event categories reported
- DCL enhancements and new system services to support auditing
- New Audit access control entry (ACE)
- New mechanism for controlling auditing messages

3.18.1.1 Distinction Between Auditing Alarms and Audits

In prior versions of the operating system, all security-related events were reported as both alarm events and audit events. Alarm events are displayed on security operator terminals only, whereas audits are recorded in the audit log file. With OpenVMS AXP Version 6.1, these events can be enabled independently as alarm events or audit events.

3.18.1.2 Auditing for Additional Security Events

OpenVMS AXP Version 6.1 extends auditing to report on the following categories of events:

- Creation, access, deaccess, and deletion of objects
- Use (or failed use) of privilege
- Changes to system parameters
- Use of identifiers as privilege
- Use of any process control system service
- Changes to the authorization databases
- Changes to system time

These extensions enable you to record more security-relevant events as they occur. You can enable auditing for these events by using the DCL command SET AUDIT. For more information, refer to the *OpenVMS Guide to System Security*.

3.18.1.3 DCL Support for Auditing

The SET AUDIT and SHOW AUDIT commands provide the management interface to the security auditing system. These commands are enhanced to support more categories of auditing, independent reporting of alarms and audits, and message flow control.

3.18.1.4 System Service Support for Auditing

To support security auditing, OpenVMS AXP Version 6.1 contains the following features:

- The $AUDIT_EVENT system service appends an event message to the system audit log file or sends an alarm to a security operator terminal.

  The operating system calls the $AUDIT_EVENT service every time a security-relevant event occurs on the system. The system service determines whether the system administrator has enabled auditing for the event by looking at the SET AUDIT settings. When the event is enabled for alarms or audits, $AUDIT_EVENT generates an audit record that identifies the process (subject) involved and lists information supplied by its caller.

- The $CHECK_PRIVILEGE system service determines whether the caller has the specified privileges or identifiers.

  The operating system calls the $CHECK_PRIVILEGE service any time a user attempts to perform a privileged function. The service performs the privilege check and looks at the SET AUDIT settings to determine whether the system administrator enabled privilege auditing. When privilege auditing is enabled, $CHECK_PRIVILEGE generates an audit record. The audit record identifies the process (subject) and privilege involved, provides the result of the privilege check, and lists supplemental event information supplied by its caller. Privilege audit records usually contain the DCL command line or the system service name associated with the privilege check.

- The $CHECK_ACCESS and $CHKPRO system services are enhanced to support auditing.

  Anytime a process (subject) attempts to access a protected object, the operating system calls the $CHKPRO system service. With OpenVMS AXP Version 6.1, the service determines whether the system administrator has enabled auditing for the associated object class event by looking at the SET AUDIT settings for the associated object class. When an alarm or an audit is required, $CHKPRO generates an audit record that identifies the process (subject) and object involved, and includes the final outcome and any supplemental event information supplied by its caller.

  The $CHECK_ACCESS system service provides a calling interface appropriate for servers and is layered on top of the $CHKPRO service. As a result, it performs object access auditing in the same manner as $CHKPRO.

For a complete description of these features, see the *OpenVMS System Services Reference Manual, Version 6.1*.

3.18.1.5 New Audit Access Control Entry

System managers can gain information on the use of individual objects. To enable alarms or audits for a specific object, system managers can use an Alarm ACE[1] or an Audit ACE, respectively.

3.18.1.6 New Mechanism for Controlling Audit Messages

OpenVMS AXP Version 6.1 limits the ability of processes to overload the system with audit messages. By restricting the number of messages, the system prevents excessive resource consumption and loss of critical audit records. The auditing subsystem monitors the total number of outstanding messages and tracks the number of messages contributed by each active process. At certain thresholds, the influx of messages is controlled by suspending processes that produce an excess number of auditable events. Message volume is controlled on a per-process basis.

3.18.2 Object Protection

The operating system now protects more classes of objects, and each one of these classes carries a consistent set of security elements. The system allows you to change these elements after the objects are created and provides a new interface to simplify object protection management.

3.18.2.1 New Classes of Objects

OpenVMS AXP Version 6.1 now protects the following object classes:

- Common event flag cluster—A set of 32 event flags that enable cooperating processes to post event notifications to each other.
- Volume—A single point of access control to all files contained on a Files–11 On-Disk Structure Level 2 disk volume.
- Resource domain—A namespace controlling access to lock management resources. OpenVMS AXP Version 6.1 protects access to locks (and their associated lock value blocks) by protecting the resource domain to which the lock resource belongs.

[1] This alarm function was formerly called ALARM_JOURNAL.

- Security class—An object class that includes all other object classes as its members. This class defines the template profiles for each object class (excluding files). Template profiles provide protection, ownership, and ACL elements for newly created objects. (Thus, the system parameters TTY_OWNER and TTY_PROT are obsolete.)

3.18.2.2 Consistent Set of Security Elements for All Object Classes

With OpenVMS AXP Version 6.1, all protected objects carry a common set of elements in a **security profile**. A security profile includes an access control list (ACL), a UIC-based protection code, and an owner UIC (User Identification Code). In previous versions, the operating system supported full security profiles for the following classes of objects: files, devices, global sections, logical name tables, and queues.

3.18.2.3 Security Profile Templates

OpenVMS AXP Version 6.1 supplies default templates for all classes of objects other than files. When you create an object, the system uses the appropriate template to construct an initial security profile for the object. You can modify any profile template to change the default values by using the DCL command SET SECURITY. For example, you can specify a default ACL for newly created mailboxes.

The security elements of files are derived in the same way as in previous versions, for example, by using existing versions and directory defaults.

3.18.2.4 Clusterwide Distribution of Security Profiles

In prior versions of the operating system, files and queues were the only object classes for which security elements, or profiles, were distributed clusterwide. With OpenVMS AXP Version 6.1, the operating system distributes security profiles clusterwide for all objects that are visible from multiple nodes in a cluster (files, disks, tapes, resource domains, volumes, queues, and security classes). Thus, if an object is accessed by two processes in a cluster on different nodes of the cluster, the system uses the same security profile for both access checks. If you change the security profile for a cluster-visible object, the system automatically distributes your changes to the other nodes in the cluster.

A new security object database stores the cluster-visible profiles for the system. This database also maintains settings during a rebooting. Each time the system starts up, it is able to retrieve object profiles from the database. This retrieval process simplifies system startup files and helps you maintain a consistent security profile across the entire cluster.

3.18.2.5 DCL Interfaces for Object Protection

You can manage all protected objects through the new DCL commands SET SECURITY and SHOW SECURITY. The SET SECURITY command modifies the security profile of an object. The SHOW SECURITY command displays the name, class, and profile of a protected object.

The SET SECURITY command supersedes the following commands:

- SET ACL
- SET FILE/PROTECTION
- SET PROTECTION
- SET VOLUME/PROTECTION

The SHOW SECURITY command supersedes the SHOW ACL command.

3.18.2.6 System Services for Object Protection

To support object protection, OpenVMS AXP Version 6.1 contains several new system services:

- The $GET_SECURITY and $SET_SECURITY system services display or modify a security profile. The $GET_SECURITY system service returns information about the security profile of a selected object. The $SET_SECURITY system service modifies the security profile of a protected object.

  $SET_SECURITY and $GET_SECURITY maintain a single master copy of a profile for every protected object in a cluster system and distribute the profile of cluster-visible objects. They also ensure that only one process at a time can modify an object's security profile.

- The $CREATE_USER_PROFILE system service returns a security profile for the specified user, modified according to the supplied item list.

- The $SET_RESOURCE_DOMAIN system service controls the association between a calling process and a resource domain. It enables a process to use the lock management system services $DEQ, $ENQ, $ENQW, $GETLKI, and $GETLKIW.

For a complete description of these features, see the *OpenVMS System Services Reference Manual, Version 6.1*.

3.18.2.7 Appropriate Access Types for Each Object Class

Each object class defines access types appropriate to that class of object. For example, files support read, write, execute, and delete access, whereas queues support read, submit, manage, and delete access. Table 3–2 defines the access types for the specified object classes.

All objects support control access, which is the ability to examine or modify a security profile.

Table 3–2 Access Types

Object Class	Access Types
Common event flag cluster	Associate, delete, control
Device	Read, write, physical, logical, control
File (including directory file)	Read, write, execute, delete, control
Group global section	Read, write, execute, control
Logical name table	Read, write, create, delete, control
Queue	Read, submit, manage, delete, control
Resource domain	Read, write, lock, control
Security class	Read, write, control
System global section	Read, write, execute, control
Volume	Read, write, create, delete, control

3.18.2.8 LIB$ Routines for Access Conversion

In OpenVMS AXP Version 6.1, each object class defines access types appropriate to that class of object. To handle conversions for object-specific access types, OpenVMS AXP Version 6.1 includes the following new library routines:

- LIB$FORMAT_SOGW_PROT

- LIB$GET_ACCNAM
- LIB$GET_ACCNAM_BY_CONTEXT
- LIB$PARSE_ACCESS_CODE
- LIB$PARSE_SOGW_PROT

OpenVMS AXP Version 6.1 still accepts and translates the traditional access names. For detailed information about the new library routines, see the *OpenVMS RTL Library (LIB$) Manual*.

3.18.2.9 Enhanced Device Protection

When mounting or initializing a disk or tape volume in OpenVMS AXP Version 6.1, the operating system checks access to the device as well as access to the volume.

To initialize or mount a volume on a device, you must have read, write, or control access to the device. System managers and users with the privilege SYSPRV have this type of access.

3.18.3 Protected Subsystems

With OpenVMS AXP Version 6.1, you can use protected subsystems to provide more sophisticated access control on objects. A **protected subsystem** is an application that, when run, causes the process running the application to be granted one or more additional identifiers. For as long as a user runs the subsystem, the user's security profile carries these additional identifiers.

Users with execute access to the application can gain access to the subsystem. Once in the subsystem, users can work with the data files and other resources of the subsystem.

To support protected subsystems, OpenVMS AXP Version 6.1 includes the following features:

- A Subsystem ACE
- A new Subsystem attribute for the DCL command SET RIGHTS_LIST and the AUTHORIZE commands GRANT/IDENTIFIER and REVOKE /IDENTIFIER
- The new DCL command qualifier /SUBSYSTEM for the MOUNT and SET VOLUME commands
- The new $SUBSYSTEM system service for saving and later restoring subsystem identifiers to a process's rights list when it exits a protected subsystem
- A PRC$M_SUBSYSTEM flag for $CREPRC (Create Process)
- A SUBSYSTEM flag for LIB$SPAWN

3.18.4 Additional Security Enhancements

Several other security enhancements have been added to OpenVMS AXP Version 6.1. These include:

- Support of facility-specific rights identifiers
- Extended use of rights identifiers
- A new ACE to control a creator's access to files in the directory of a resource identifier

- New privileges
- System service changes to maintain SYSUAF.DAT context

3.18.4.1 Facility-Specific Rights Identifiers

OpenVMS AXP Version 6.1 supports facility-specific rights identifiers. These types of rights identifiers allow a system manager to give users rights without assigning user privileges.

During the product's installation, facility identifiers combine an application's facility code with the value specified by that application. The application then uses the $ADD_IDENT service to register the application's facility identifiers with the operating system.

For more information about the use of facility-rights identifiers, refer to the *OpenVMS Programming Concepts Manual*.

3.18.4.2 Rights Identifier Attributes

OpenVMS AXP Version 6.1 extends the use of rights identifiers with the following identifier attributes:

Holder Hidden	Prevents other people from getting a list of users who hold an identifier unless they own the identifier themselves.
Name Hidden	Allows holders of an identifier to have it translated (either from binary to ASCII or from ASCII to binary) but prevents unauthorized users from translating the identifier.
No Access	Makes any access rights of the identifier null and void. This attribute is intended as a modifier for a resource identifier or the Subsystem attribute.
Subsystem	Allows holders of the identifier to create and maintain protected subsystems by assigning the Subsystem ACE to the application images in the subsystem.

A complete list of identifier attributes is in the *OpenVMS Guide to System Security*.

3.18.4.3 Greater Control of Certain File Profiles

A process without system privileges can create a file in a directory owned by a resource identifier. In previous versions of the operating system, a file created under these conditions was assigned an extra ACE with a combination of control access and ownership access. The purpose of the extra ACE was to guarantee access to the file by its creator. However, depending on the explicit security controls set up in the directory, the extra ACE can interfere with the accessibility of the files. A new Creator ACE provides the tool to solve these problems.

In OpenVMS AXP Version 6.1, you can assign a Creator ACE to the parent directory to determine the access specified by the extra ACE. You can also use a Creator ACE to suppress the extra ACE. If there is a Creator ACE in the ACL for the parent directory, the system propagates the access specified in the Creator ACE to the new ACE. If a directory lacks a Creator ACE, the system adds an extra ACE, which gives you control access plus the access specified in the owner field of the file's protection code. A Creator ACE with ACCESS=NONE specified suppresses the addition of the extra ACE.

3.18.4.4 New Privileges

Four new privileges have been added to OpenVMS AXP Version 6.1: AUDIT, IMPORT, UPGRADE, and DOWNGRADE.

The AUDIT privilege allows software programs to append audit records to the system security audit log file using one of four system services: $AUDIT_EVENT, $CHECK_PRIVILEGE, $CHKPRO, or $CHECK_ACCESS.

The three other new privileges—IMPORT, UPGRADE, and DOWNGRADE—specifically support SEVMS, a product that enhances the security of OpenVMS.

3.18.4.5 Setting Context Within SYSUAF.DAT

The $CHECK_ACCESS, $GETUAI, and $SETUAI system services have a new **contxt** argument to maintain context within the system user authorization file (SYSUAF.DAT).

3.18.5 Security Management Changes

In addition to changes described in this chapter, security managers should take note of the following:

- Modification of the rights list (SYS$SYSTEM:RIGHTSLIST.DAT) to deny world read access
- Changes to the access control list editor (ACL editor) to support new object classes, new access control entries (ACEs), new access types, and new identifier attributes
- Additional auditing event categories and selection criteria for the Audit Analysis utility (ANALYZE/AUDIT)
- Changes to audit server startup (see *OpenVMS System Manager's Manual*)
- Enhancements to the Authorize utility and the DCL command SET RIGHTS_LIST to support new identifier attributes
- Support for protected subsystems, which includes modifications to the Mount utility (MOUNT) and the DCL command SET VOLUME
- New system parameters:
  - LGI_CALLOUTS and NET_CALLOUTS provide support for LOGINOUT routines
  - SECURITY_POLICY allows sites to control particular pieces of operating system functionality (see online help for the SYSGEN utility)

3.19 System Parameters

This section lists new system parameters for OpenVMS AXP Version 6.1.

3.19.1 New System Parameters for OpenVMS AXP

Table 3–3 describes new system parameters for OpenVMS AXP Version 6.1. For a listing of all system parameters, see the *OpenVMS System Management Utilities Reference Manual*.

Table 3–3 New System Parameters for OpenVMS AXP Version 6.1

Parameter	Description
DBGTK_SCRATCH	Specifies how many pages of memory is allocated for the remote debugger. This memory is allocated ONLY if remote debugging is enabled with the 8000 boot flag (described below). Normally, the default value is adequate, but if the remote debugger issues an error message, you should increase this value. See the *OpenVMS AXP Device Support: Developer's Guide* for more information.
GH_EXEC_CODE	Specifies the size in pages of the execlet code granularity hint region.
GH_EXEC_DATA	Specifies the size in pages of the execlet data granularity hint region.
GH_RES_CODE	Specifies the size in pages of the resident image code granularity hint region.
GH_RES_DATA	Specifies the size in pages of the resident image data granularity hint region.
IMGREG_PAGES	Specifies the number of pages to reserve in P1 space for images to be installed with shareable address data. If IMGREG_PAGES is set to 0, no images are installed with shared address data. The default is 600. For more information, see the INSTALL section in the *OpenVMS System Manager's Manual*.
PFN_COLOR_COUNT	Specifies the number of buckets (colors) into which all members of the zeroed page list and all unencumbered members of the free page list are sorted. OpenVMS AXP systems may derive a preferred page color from a request to map a given virtual page and attempt to map that virtual page to a PFN of matching "color." This results in less variance in which cache blocks may be used when accessing that page. This may or may not improve performance, depending upon the application. Note that when you increase this parameter, you should also increase the ZERO_LIST_HI parameter. PFN_COLOR_COUNT is a special system parameter.
SHADOW_SYS_TMO	This parameter has the following two distinct usages: • Use this parameter at system boot time, when this is the FIRST node in the cluster to boot and to create this specific shadow set. If the proposed shadow set is not currently mounted in the cluster, use this parameter to extend the time a booting system waits for all former members of the shadowed system disk to become available. • Use this parameter once the system successfully mounts the virtual unit and begins normal operations. In this usage the SHADOW_SYS_TMO parameter controls the time the operating system waits for errant members of a system disk. (Use the SHADOW_MBR_TMO parameter to control the time the operating system waits for the errant members of an application disk). This parameter applies only to members of the system disk shadow set. All nodes using a particular system disk shadow set should have their SHADOW_SYS_TMO parameter set to the same value once normal operations begin. The default value is 20 seconds. Change this parameter to a higher value if you want the system to wait more than the 20-second default for all members to join the shadow set. You can set the parameter value to a range of 20 through 65,535 seconds.
SHADOW_SYS_WAIT	Extends the time a booting system waits for all current members of a mounted shadowed system disk to become available to THIS node. The shadow set must already be mounted by at least one other cluster node for this parameter to take effect. The default value is 256 seconds. Change this parameter to a higher value if you want the system to wait more than the 256-second default for all members to join the shadow set. You can set the parameter value to a range of 1 through 65,535 seconds.

(continued on next page)

Table 3–3 (Cont.) New System Parameters for OpenVMS AXP Version 6.1

Parameter	Description
SYSTEM_CHECK	Used to investigate intermittent system failures by enabling a number of runtime consistency checks on system operation and recording some trace information. Enabling SYSTEM_CHECK causes the system to behave as if the following system parameter values are set (note that the values of the following parameters are not actually changed): **Parameter / Value / Description** BUGCHECKFATAL / 1 / Crash the system on nonfatal bugchecks POOLCHECK / %X616400FF / Enable all poolchecking, with an allocated pool pattern of %x61616161 ('aaaa') and deallocated pool pattern of x64646464 ('dddd') MULTIPROCESSING / 2 / Enable full synchronization checking While SYSTEM_CHECK is enabled, the previous settings of the BUGCHECKFATAL and MULTIPROCESSING parameters are ignored. However, setting the parameter POOLCHECK to a nonzero value overrides the setting imposed by SYSTEM_CHECK. Setting SYSTEM_CHECK causes certain image files that are capable of the additional system monitoring. These image files are located in SYS$LOADABLE_IMAGES and can be identified by the suffix _MON.
XQPCTL2	Controls improved concurrency. The default value of XQPCTL2 is 0, which turns off improved concurrency. To turn on improved concurrency, set XQPCTL2 to 1. XQPCTL2 is a special system parameter.
XQPCTLD1	Controls multithreading, which can only be used by PATHWORKS servers. The default value of XQPCTLD1 is 0, which disables multithreading. To enable multithreading, set XQPCTLD1 to 8. XQPCTLD1 is a special system parameter.

3.19.2 New System Parameters Previously Supported by OpenVMS VAX

Table 3–4 describes system parameters, previously supported on OpenVMS VAX systems, that are now also supported on OpenVMS AXP Version 6.1.

Table 3–4 System Parameters Previously Supported on OpenVMS VAX

Parameter	Description
SECURITY_POLICY	Allows a system to run in a C2 or B1 configuration and subset out particular functions that are outside of the evaluated configuration or to preserve compatibility with previous versions of the operating system. See the *OpenVMS Guide to System Security* for further information on the C2 and B1 evaluated configurations.

(continued on next page)

Table 3–4 (Cont.) System Parameters Previously Supported on OpenVMS VAX

Parameter	Description
SHADOWING	Enables or disables shadowing and specifies the mode of shadowing operations that you want to enable. SHADOWING is a value that specifies the type of disk class driver that is loaded on the system: DUDRIVER, DSDRIVER, or SHDRIVER. Specify one of the following values:

Value	Description
0	No shadowing is enabled; DUDRIVER is used. This is the default value.
2	Phase II shadowing; SHDRIVER and DUDRIVER are used. Phase II shadowing provides shadowing of all MSCP compliant DSA disks located anywhere on a standalone system or a VMScluster system.

	See the *Volume Shadowing for OpenVMS* for more information about setting system parameters for volume shadowing.
SHADOW_MAX_COPY	Controls how many parallel copy threads are allowed on a given node. The value of SHADOW_MAX_COPY can range from 0 to 42. The default value is 4. Carefully consider the needs of each shadowed system when setting this parameter. For example, the default value of 4 might be too high for a small satellite node and too low for a node with substantial CPU power. Too high a value for SHADOW_MAX_COPY can affect performance by allowing too many copy threads to operate in parallel. Too low a value unnecessarily restricts the number of threads your system can effectively handle. See *Volume Shadowing for OpenVMS* for more information about setting system parameters for volume shadowing.
SHADOW_MBR_TMO	Controls the amount of time the system tries to fail over physical members of a shadow set before removing them from the set. The SHADOW_MBR_TMO parameter replaces the temporary VMSD3 parameter used in prior releases. The SHADOW_MBR_TMO parameter is valid only for use with OpenVMS Volume Shadowing (Phase II). Use the SHADOW_MBR_TMO parameter to specify the number of seconds, in decimal from 1 to 65,535, during which recovery of a repairable shadow set is attempted. If you do not specify a value, or if you specify 0, the default delay of 20 seconds is used. Because SHADOW_MBR_TMO is a dynamic parameter, you should use the SYSGEN command WRITE CURRENT to permanently change its value. To temporarily change the value of SHADOW_MBR_TMO on a running system, you must use the SYSGEN command WRITE ACTIVE.
SHADOW_SYS_DISK	Enables (1) or disables (0) Phase II shadowing only. Disable the SHADOW_ SYS_DISK parameter if the boot device is not a member of a Phase II shadow set. The default value is disabled. Enable this parameter if you want to shadow the boot device. All nodes booting off a particular system disk shadow set should set their SHADOW_SYS_ DISK parameter to 1. See *Volume Shadowing for OpenVMS* for more information about setting system parameters for volume shadowing.
SHADOW_SYS_UNIT	An integer value that contains the virtual unit number of the system disk. The default value is 0. The maximum value allowed is 9999. This parameter is effective only when the SHADOW_SYS_DISK parameter has a value of 1. This parameter should be set to the same value on all nodes booting off a particular system disk shadow set. See *Volume Shadowing for OpenVMS* for more information about setting system parameters for volume shadowing.

3.20 New SYSMAN Commands: IO SET EXCLUDE and IO SHOW EXCLUDE

A new feature in SYSMAN allows you to permanently exclude certain devices from autoconfiguration.

Some SCSI devices—especially optical devices—do not fully follow the SCSI standard. As a result, the system autoconfigures these devices as disks. This autoconfiguration prevents you from using an alternate class driver to manage the device. Excluding the device as a disk allows you to configure it later in startup (in SYCONFIG.COM).

Note that autoconfiguration is not a problem on standalone systems on which shadowing is not running, because you can use the IO AUTOCONFIGURE /EXCLUDE command. In a cluster, or with shadowing turned on, you must configure all devices that might be disks much earlier in the boot process—well before SYCONFIG.COM is invoked. However, you can use the new permanent exclusion list created with IO SET EXCLUDE in both clustered and nonclustered systems, and with or without shadowing.

3.20.1 IO SET EXCLUDE

Sets the permanent exclusion list to be used when configuring devices automatically.

Format

IO SET EXCLUDE = (device_name)

Parameter

device_name
Specifies the device type to be excluded from automatic configuration. Use valid device names or mnemonics that indicate the devices to be included in the permanent exclusion list. You can specify wildcards.

Description

Sets the permanent exclusion list to be used when configuring devices.

Example

```
SYSMAN> IO SET EXCLUDE=(DKC500,DKD*)
```

This example specifies that DKC500 and all DKD devices are not to be autoconfigured.

Refer to the /SELECT qualifier of the SYSMAN command IO AUTOCONFIGURE in the *OpenVMS System Management Utilities Reference Manual* for additional examples that show how to specify device names.

3.20.2 IO SHOW EXCLUDE

Displays the permanent exclusion list used in the autoconfiguration of devices.

Format

IO SHOW EXCLUDE

Description

The IO SHOW EXCLUDE command displays the permanent exclusion list on the console. This list is used in the autoconfiguration of devices.

Example

```
SYSMAN>  IO SHOW EXCLUDE
%SYSMAN-I-IOEXCLUDE, the current permanent exclusion list is: DKC500,DKD*
```

This example shows the permanent exclusion list used in the autoconfiguration of devices; the current list contains DKC500 and all DKD devices.

3.21 LAT New Features

The OpenVMS AXP Version 6.1 operating system includes new features that have been added to the SET HOST/LAT command, the LAT Control Program (LATCP) utility, and the programming environment (new LAT item codes). In addition to the summaries provided in this section, see the following for more detailed information:

- *OpenVMS System Manager's Manual*, for LAT system management information
- *OpenVMS System Management Utilities Reference Manual*, for new LATCP commands and qualifiers
- *OpenVMS DCL Dictionary*, for information about the SET HOST /LAT command
- Section 4.22, for information about new LAT programming features

3.21.1 New Features of SET HOST /LAT Command

This section describes new tasks you can perform at the DCL level by using the SET HOST /LAT command.

3.21.1.1 Intrasystem LAT Connections

The LAT software now provides the ability to make an intrasystem LAT connection. For example, if the local node is offering service FOO and no other node is offering FOO, then the following command will create an intrasystem LAT connection:

```
$ SET HOST /LAT FOO
```

Note that the LAT port driver (LTDRIVER) and the LAT Ancillary Control Process (LATACP) will also consider the local node as a candidate for an intrasystem connection if it is the highest rated node offering the selected service.

Note

The SET HOST /LAT 0 command will *not* create an intrasystem LAT connection (like SET HOST 0 does) because 0 is used by the DECnet software for other purposes, making its use in this context an illegal DECnet node name. (Because a node on your local area network (LAN) can offer a LAT service called 0, the SET HOST /LAT 0 command is valid.)

3.21.1.2 Propagation of Dialup Bit

If a user logs in to an account from a dialin source, and the user issues a SET HOST /LAT command, the LTPAD program will propagate the dialup characteristic to the remote node.

3.21.1.3 Using the SET HOST /LAT in a Command Procedure

You can now use the SET HOST /LAT in a command procedure (.COM file). However, note the following:

- Digital does *not* recommend that you use SET HOST /LAT in a command procedure if you intend to include user names and passwords as well. That practice can lead to a security breach.
- This feature has been tested between OpenVMS systems and other operating systems but works best when the remote system is another OpenVMS system. Using SET HOST /LAT in a command procedure with systems other than OpenVMS might produce unexpected results because of differing implementations of type-ahead buffers on the other systems.

  For example, using SET HOST /LAT from an OpenVMS system in a command procedure to a remote ULTRIX node produces unexpected results. Using the DECnet SET HOST command in the same environment produces unexpected results as well.

3.21.1.4 New Qualifiers to SET HOST /LAT Command

New SET HOST /LAT qualifiers include the following:

- /AUTOPROMPT (default) and /NOAUTOPROMPT

  Specifying this qualifier determines whether the Username: prompt displays automatically on the screen when a SET HOST /LAT command is entered.
- /QUEUE and /NOQUEUE (default)

  Specifying this qualifier determines whether a connect request to a service already in use will be queued (if that target service has queuing enabled).

See Section 2.1.16 and the *OpenVMS DCL Dictionary* for more detailed information about these new qualifiers to the SET HOST /LAT command.

3.21.2 New Features of the LAT Control Program (LATCP) Utility

The following sections describe enhancements made to the LAT software, which you can implement and manage using the LAT Control Program (LATCP) utility.

3.21.2.1 New LATCP Commands and Qualifiers

Table 3–5 and Table 3–6 list new commands and qualifiers for the LAT Control Program (LATCP) utility. See the *OpenVMS System Management Utilities Reference Manual* for complete descriptions.

Table 3–5 New LATCP Commands

Command	Function
DELETE QUEUE_ENTRY	Deletes an incoming queued request from the local node.
SHOW QUEUE_ENTRY	Displays information about requests, or entries, queued on the local node.
SCROLL	Allows you to retrieve information that has scrolled off the screen.

Table 3–6 New LATCP Command Qualifiers

New Qualifier	Commands	Function
/CONNECTIONS /NOCONNECTIONS	SET SERVICE	Allows you to specify whether a service offered by an OpenVMS system accepts incoming connections.
/COUNTERS	SHOW PORT	Allows you to display the counters kept for the port.
/LIMITED	CREATE PORT SET PORT	Allows you to specify that a logical port on your local node is limited to a service in the same way a port created using the /DEDICATED qualifier is dedicated to an application service. The difference is that ports created using the /LIMITED qualifier are under the control of the system login image (LOGINOUT.EXE) instead of an application program.
/LIMITED	CREATE SERVICE SET SERVICE	Used in conjunction with the SET PORT /LIMITED and CREATE PORT /LIMITED commands. Allows you to specify that a service is a limited service, using devices assigned the limited characteristic and associated with (mapped to) this limited service.
/PORT=port-name	ZERO COUNTERS	Allows you to specify the port (on your local node) for which you want counters reset.
/QUEUED /NOQUEUED	SET SERVICE	Allows you to specify whether a locally offered limited (/LIMITED) or application (/DEDICATED) service is allowed to have queued connections when all ports are busy.
/STATUS	SHOW NODE	Allows you to display information about the local node, such as the number of active circuits, sessions, and incoming queue entries.

3.21.2.2 Creating Limited Services

Application services with dedicated ports allow you to create a predetermined number of LAT devices (for example, LAT terminals, which the system identifies with the LTAn prefix in system displays) that are under the control of a process supplied by the system. In that environment, however, the user cannot log in to the service because there is no way for dedicated LTA devices to run the system login image (LOGINOUT.EXE).

On Version 6.1 of the OpenVMS AXP operating system, you can now create a **limited** service that allows users to log in to a predetermined number of LTA devices associated with that limited service. Creating a limited service in this way allows you to control the number of LAT users on your system.

See the *OpenVMS System Manager's Manual* for more information about creating limited services.

3.21.2.3 Queuing Incoming Requests

If you attempt to connect to a limited or application service, the LAT software will (by default) queue, rather than reject, this connection request if the service has no available ports.

See the *OpenVMS System Manager's Manual* for more information about how to set up your system to queue incoming requests.

3.21.2.4 Rating Modifications

Beginning with Version 6.1 of the OpenVMS AXP operating system, limited and application services no longer rely exclusively on the dynamically calculated service rating. Instead, they use a portion of the dynamic rating based on how many ports are available for the service. This rating procedure for limited and application services on OpenVMS AXP systems follows the terminal server rating algorithm for services and available ports that the service offers, while at the same time takes into account the availability of the AXP node (which is the factor used to calculate the dynamic rating).

For more information, see the description of the /STATIC_RATING qualifier for the SET PORT and SET SERVICE commands in the *OpenVMS System Management Utilities Reference Manual*.

3.21.2.5 Controlling Logical Name Translations

The LAT Control Program (LATCP) utility has been modified to allow you to prevent it from translating logical names. In the following example, LATCP creates a service name called SOME_NAME, translates FOO as a logical name to SOME_NAME, and then uses that translation SOME_NAME.

```
$ DEFINE /SYSTEM FOO SOME_NAME
$ LCP :== $LATCP
$ LCP CREATE SERVICE FOO
```

In this example, you can override this translation by using the following commands:

```
$ DEFINE /SYSTEM FOO SOME_NAME
$ LCP :== $LATCP
$ LCP CREATE SERVICE _FOO
```

When a name is preceded by an underscore (_), LATCP removes the underscore and uses the remaining name as is. In the previous example, then, LATCP creates a service called FOO (not _FOO) and does not translate the logical name to SOME_NAME.

Note

This change might affect current commands. Examine your LAT$SYSTARTUP.COM file to make sure that you are not already creating names with leading underscores. For example, you might have created a service name called _FOO by using the following commands:

```
$ DEFINE /SYSTEM FOO SOME_NAME
$ LCP :== $LATCP
$ LCP CREATE SERVICE _FOO
```

If you want to retain the name _FOO, you must add a second underscore (because LATCP now removes the first leading underscore). For example:

```
$ LCP CREATE SERVICE __FOO
```

After LATCP removes the first underscore, the second underscore indicates that no logical name translation is required. LATCP will then create a service named _FOO.

3.21.2.6 Specifying Wildcards

The LAT Control Program (LATCP) utility now allows you to specify any valid wildcard when you enter a SHOW NODE or SHOW SERVICE command. For example, when you enter the following command, the system will display the status and characteristics of all nodes that begin with the letter A:

```
LATCP> SHOW NODE A*
```

4

Programming Features

This chapter highlights new programming features for OpenVMS AXP Version 6.1. For detailed information about a feature, refer to the identified manuals in the OpenVMS documentation set. In addition, refer to the *OpenVMS AXP Version 6.1 Release Notes* for information about possible system impact.

4.1 Device Support on OpenVMS AXP Systems

OpenVMS AXP Version 6.1 introduces formal support for user-written device drivers and a new device driver interface known as the Step 2 driver interface. The Step 2 driver interface replaces the temporary Step 1 driver interface that was provided in OpenVMS AXP Versions 1.0 and 1.5.

The major feature of the new Step 2 driver interface is support for writing device drivers in the C programming language. An entire device driver can now be written in C. An example device driver written in C is provided in the SYS$EXAMPLES directory. For more information about writing device drivers in C, see the *OpenVMS AXP Device Support: Developer's Guide*.

Note

To write a Step 2 driver in C, the optional DEC C compiler for OpenVMS AXP is required.

Step 2 drivers can continue to be written in MACRO-32. However, source code changes are required to all Step 1 drivers to bring them into compliance with the Step 2 interfaces. Step 1 drivers are not loadable starting with OpenVMS AXP Version 6.1. *Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver* describes the changes needed to convert a Step 1 driver written in MACRO-32 to a Step 2 driver. To find out how to convert an OpenVMS VAX device driver to a Step 2 driver, see *Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver*.

A fourth new book, *OpenVMS AXP Device Support: Reference*, provides reference material applicable to device drivers written in C and MACRO-32.

4.2 OpenVMS AXP System-Code Debugger

OpenVMS AXP Version 6.1 supports a new programming tool that can be used to debug nonpageable system code and device drivers running at any IPL. The OpenVMS AXP System-Code Debugger (system-code debugger) lets you use the familiar OpenVMS Debugger interface to observe and manipulate system code interactively as it executes.

Using the system-code debugger, you can perform the following tasks:

- Control the system software's execution—-stop at points of interest, resume execution, intercept fatal exceptions, and so on.

- Trace the execution path of the system software
- Monitor exception conditions
- Examine and modify the values of variables
- In some cases, test the effect of modifications without having to edit the source code, recompile, and relink

The system-code debugger is a symbolic debugger. You can specify variable names, routine names, and so on, precisely as they appear in your source code. The system-code debugger can also display the source code where the software is executing and allow you to step by source line.

You can use the system-code debugger with the following languages:

- C
- Bliss
- Macro

Note

The Bliss compiler is not available as a product for OpenVMS AXP.

The system-code debugger recognizes the syntax, data typing, operators, expressions, scoping rules, and other constructs of a given language. If your program is written in more than one language, you can change the debugging context from one language to another during a debugging session.

For information about how to use the system-code debugger and how it differs from the OpenVMS Debugger, see the *OpenVMS AXP Device Support: Developer's Guide*. For information about OpenVMS Debugger commands, see the *OpenVMS Debugger Manual*.

4.3 OpenVMS Debugger New Features

This section describes the following OpenVMS Debugger (debugger) features, which are new in this release:

- Language support
- Floating-point arithmetic enhancements
- DECwindows Motif user interface enhancements
- Support for debugging resident images
- Static watchpoint support
- DCL DEBUG command enhancements
- CONNECT/DISCONNECT command support
- OpenVMS AXP System-Code Debugger support
- Online help changes

4.3.1 Language Support

On OpenVMS AXP systems, the debugger now provides support for Basic and PL/I languages. For more detailed information, see the *OpenVMS Debugger Manual*.

4.3.2 Floating-Point Arithmetic Enhancements

The following enhancements have been made to floating-point data type support:

- The support for IEEE floating-point data types is designed to use high-performance instructions. This means that debug expression evaluation involving IEEE arithmetic operations can produce infinities and NaNs, but not denormals. For more information on OpenVMS AXP high-performance IEEE support, see the *Alpha Architecture Manual*. For more information on IEEE floating-point data types, see the *IEEE Floating-Point Standard (ANSI/IEEE 754-1984)*.
- Support for the IEEE extended floating-point data type has been added. This is a 128-bit (16 byte) type. DEC FORTRAN is the only language with extended floating-point support. Table 4–1 describes the debugger commands that accept qualifiers or parameters associated with the extended floating-point data type.

Table 4–1 IEEE Extended Floating-Point Data Type Qualifiers and Parameters

Debugger Command	New Parameters	New Qualifiers
EXAMINE	None	/EXTENDED_FLOAT (/X_FLOAT for short)
DEPOSIT	None	/EXTENDED_FLOAT (/X_FLOAT for short)
MONITOR	None	/EXTENDED_FLOAT (/X_FLOAT disallowed)
SET TYPE	EXTENDED_ FLOAT or X_ FLOAT	None

For more detailed information, see the *OpenVMS Debugger Manual*.

4.3.3 DECwindows Motif User Interface Enhancements

This version of the debugger offers the following additional features in the DECwindows Motif interface.

- Single window design for most common operations

  The main window now includes the former source and control panel windows. A single menu bar contains the pulldown menus for this window.
- Single window design for Breakpoint, Monitor, Register, and Tasking Views

  These views no longer appear within the control panel window but within their own optional view window. The Instruction View still appears in its own window with a separate menu bar.
- Additional and adjusted pulldown menus and menu items

  These include a new Commands pulldown menu on the main window and Monitor and Register pulldown menus on the optional view window. The Options menu on the main window no longer lists individual views but brings up a dialog box from which views can be chosen.

- Editable source in a new window

  The editor is accessed from the Edit File item in the Commands pulldown menu on the main window.

- Context-sensitive popups

  Context-sensitive popup menus can be accessed by positioning your mouse pointer in the debugger window panes and clicking MB2. Each popup allows you to issue commands and choose functions from a list of those available for the pane in which your mouse pointer is located.

- User overrides for radix and type information in Monitor and Register Views

  This feature was formerly available only through the command line.

- Ability to expand and modify breakpoints in the Breakpoint View

  The ability to modify breakpoints was formerly available only in the source display or command line. Breakpoint entries can now be expanded to show conditions and actions associated with the breakpoint.

- Enhanced pushbutton customization

  The design of the Customize dialog box has been simplified. In addition, the ability to resequence the push buttons in the main window has been added.

- Keypad support

  The following commands have been mapped to individual keys on the computer keypad:

Step/Line	KP0
Examine	KP1
Go	KPcomma
Step/Into	GOLD-KP0
Step/Over	BLUE-KP0
Examine^	GOLD-KP1
Show Calls	KP5
Show Calls 3	GOLD-KP5

  To issue one of these commands, press the key indicated, followed by the Enter key on the keypad.

  You can change the commands represented by each key, or map commands to other keys on your keyboard, by customizing the EnterCmdOnCmdLine entry in the debugger resource file.

  If you are mapping to other keys, you also need to consult the key designations listed in the KeySym Encoding chapter of the *X and Motif Quick Reference Guide*.

4.3.4 Support for Debugging Resident Images

The debugger now provides support for debugging resident images, a feature of the OpenVMS AXP operating system.

Resident images are shareable or executable images that have the following qualities:

- Linked with the /SHARE and /SECTION_BINDING=CODE linker qualifiers
- Installed as resident images using the Install utility

This feature lets the operating system split, or slice, the contents of images and sort the pieces so that they can be placed with other pieces that have the same page protection in the same area of memory. Slicing images leads to improvements in performance and access.

4.3.5 Static Watchpoint Support

In previous versions, the OpenVMS AXP operating system did not allow the debugger to intercept system service calls. This meant that the debugger treated all watchpoints as nonstatic watchpoints, because some system services could not accept the static watchpoint algorithm. This algorithm includes write-protecting the page of memory in which the watched variable is stored.

In this version, the operating system does allow the debugger to intercept system service calls. This means that the debugger can support the faster static watchpoint algorithm for static variables.

System service interception is on by default, but you can disable interception prior to a debugging session by issuing the following command:

```
$  DEFINE SSI$AUTO_ACTIVATE OFF
```

To re-enable system service interception, issue one of the following commands:

```
$  DEFINE SSI$AUTO_ACTIVATE ON
$  DEASSIGN SSI$AUTO_ACTIVATE
```

4.3.6 DCL DEBUG Command Enhancements

Two new qualifiers have been added to the DCL DEBUG command: /KEEP and /RESUME.

Use the /KEEP qualifier to invoke the Kept Debugger at the DCL prompt. For example:

```
$ DEBUG/KEEP
```

Use the DEBUG/KEEP command instead of the following command:

```
$ RUN SYS$SHARE:DEBUGSHR
```

Use the /RESUME qualifier to resume your debugging session after executing Ctrl/Y to interrupt the character-cell debugger. For example:

```
$ DEBUG/RESUME
```

The DEBUG/RESUME command is equivalent to the previously accepted command:

```
$ DEBUG
```

Note that the following key sequence commands cannot be used with the debugger (these command sequences will cause internal debugger errors):

- RUN program-name/DEBUG, Ctrl/Y, and DEBUG/KEEP
- DEBUG/KEEP, Ctrl/Y, and DEBUG/RESUME

4.3.7 CONNECT/DISCONNECT Command Support

An enhanced CONNECT command and a new DISCONNECT command allow you to debug multiprocessing programs that conform to a client/server model. In this model, the user, or client, initiates connection to the server, which processes client requests. The server spawns threads to handle programming tasks. This debugger requires that the client/server model be implemented on a single machine.

The tasks required to execute a debugging session in the client/server multiprocessing model are described in the following list. Remember to issue all debugger multiprocessing commands, including the command to create your debugging process, from the debugger's command-line interface or command-entry prompt. Multiprocessing commands do not have mouse action equivalents.

1. Establish a multiprocessing debugging configuration by defining DBG$PROCESS MULTIPROCESS. Use the /GROUP qualifier to specify that both processes were created within the same UIC group:

   ```
   $ DEFINE/GROUP DBG$PROCESS MULTIPROCESS
   ```

2. Invoke the debugger:

   ```
   $ DEBUG/KEEP
   ```

3. Issue the CONNECT command with a process specification parameter to interrupt an image running without debugger control. For example:

   ```
   DBG> CONNECT JONES_3
   ```

4. Use any of the debugger's multiprocess-related commands to manipulate, examine, or broadcast other debugger commands to processes.

5. When you finish debugging your program, enter the DISCONNECT command with a process specification as shown in the following example:

   ```
   DBG> DISCONNECT JONES_3
   ```

   This command releases the process from debugger control without terminating the process.

   Unexpected results can occur if you omit the DISCONNECT command and then exit the debugger. A process initiated without debugger control, to which you have then connected the debugger, continues running even after you have exited the debugging session. Until you issue an explicit DISCONNECT command, this process continues to function as if a connection still exists, and may attempt to complete tasks and transfer data.

6. Enter EXIT or QUIT when you want to end the entire debugging session.

For more detailed information, see the *OpenVMS Debugger Manual*.

4.3.8 OpenVMS AXP System-Code Debugger Support

On AXP systems, if you have invoked the debugger with the DEBUG/KEEP command, you can use the CONNECT and REBOOT commands to debug AXP operating system code. These commands are designed to work in conjunction with the OpenVMS AXP System-Code Debugger, a kernel debugger that you activate through the OpenVMS Debugger.

Typically, you issue the CONNECT command from a timesharing AXP machine (running the debugger) to connect to a standalone AXP machine (running the OpenVMS AXP operating system code). You issue the REBOOT command to execute or re-execute the program containing the operating system code and to reboot the system.

In some cases, you may find that you need to use the less robust Delta/XDelta Debugger to debug operating system code (for example, when you have access to only one AXP machine for debugging), but in general, the OpenVMS AXP System-Code Debugger is the preferred debugger for debugging OpenVMS AXP operating system code.

For information on the syntax of the CONNECT and REBOOT commands, see online help for each command.

For information on activating the OpenVMS AXP System-Code Debugger from the OpenVMS Debugger and debugging operating system code, see the *OpenVMS AXP Device Support: Developer's Guide* manual.

4.3.9 Online Help Changes

Online help has been enhanced on OpenVMS AXP Version 6.1 to contain full reference material. This material has been removed from the *OpenVMS Debugger Manual*.

In addition, online help is available through the following OpenVMS printable files in both .TXT and .PS format:

SYS$HELP:DBG$HELP.PS
SYS$HELP:DBG$HELP.TXT

4.4 OpenVMS RMS Reverse-Search Key Option

For random, keyed access to an index file, OpenVMS RMS returns records with keys that have the same value as the specified search key, unless otherwise directed. Prior to this release, RMS provided two search key options:

- RAB$M_EQNXT (RAB$M_KGE). This option returns records that have the same key value as the specified search key, or the next greater key value, according to the sort order for the current key of reference.
- RAB$M_NXT (RAB$M_KGT). This option returns records that have the next greater key value, according to the sort order for the current key of reference.

OpenVMS VAX Version 6.1 and OpenVMS AXP Version 6.1 introduce a search key option that reverses the direction of the search when used with the two existing search key options. When you specify the reverse-key (RAB$M_REV) option, RMS returns the previous record (next lower key) rather than the next record (next greater key). You can use the following combinations:

- RAB$M_REV + RAB$M_EQNXT. This combination of search key options returns records that have the same key value as the specified search key, or the next lower key value, according to the sort order for the current key of reference.
- RAB$M_REV + RAB$M_NXT. This combination of search key options returns records that have the next lower key value, according to the sort order for the current key of reference.

In all cases, you specify search key options symbolically using the mask bits in the RAB$L_ROP field of the record access block (RAB) structure. Table 4–2 identifies the results of all combinations of the REV, EQNXT, and NXT bits.

Table 4–2 Keyed Search Combinations

Search Result	REV	EQNXT	NXT
Equal (default)	0	0	0
Greater-than or equal	0	1	0
Greater-than (next)	0	0	1
RMS$_ROP error	0	1	1
RMS$_ROP error	1	0	0
Less-than or equal	1	1	0
Less than (previous)	1	0	1
RMS$_ROP error	1	1	1

You should consider the following design characteristics when choosing the reverse-search key option:

- The reverse-search key option is restricted to keyed access to indexed files.
- The reverse-search key option reverses the search direction for random keyed access *only*. Sequential access still proceeds forward *only*, according to the sort order.
- When the previous record is one of a chain of duplicates, RMS returns the first record in the chain.
- Using the reverse-search option to retrieve successive records with duplicate keys requires additional compiler or application program logic.
- On-disk data structures are designed to provide optimum performance for forward searches. Reverse search performance may be diminished, especially for applications that process long chains of deleted records.
- A performance caching optimization has been built into the reverse-search option to improve performance when retrieving successive previous records. To take advantage of this feature, specify full key sizes (RAB$B_KSZ) and select the RAB$M_NXT search option.

For more detailed information, see the *OpenVMS Record Management Services Reference Manual* and the *Guide to OpenVMS File Applications*.

4.5 Convert/Reclaim Utility Qualifiers

The DCL interface to the Convert/Reclaim utility has been enhanced.

4.5.1 New /KEY Qualifier

The /KEY qualifier to the DCL command CONVERT/RECLAIM lets you reclaim index file buckets for a specified key(s). If you also request statistics, they are reported separately for each specified key. Prior to this release, only key 0 index and data buckets were reclaimable.

Use the following format:

```
CONVERT/RECLAIM/KEY=(key_number[,...]) filename
```

The default is to reclaim all bucket space and, if statistics are requested, to provide a single summary report.

4.5.2 Enhancement of Display from CONVERT/RECLAIM/STATISTICS Command

The display format for CONVERT/RECLAIM/STATISTICS command now includes CPU and I/O statistics, similar to the display format for the CONVERT /STATISTICS command.

4.6 CONV$RECLAIM Key Selection Parameter

The CONV$RECLAIM key selection parameter has been added.

The callable Convert/Reclaim utility routine CONV$RECLAIM now provides an optional parameter, *key_number*, for selectively reclaiming buckets by key. If this parameter is omitted, or if the routine is passed a NULL value, reclamation is done for all keys in the file as in previous versions. If a valid key number for the file is passed in this parameter, the reclamation is limited to the index and data buckets for the specified key.

The following calling format includes the key selection parameter:

```
CONV$RECLAIM input_filespec [,statistics_blk][,flags][,key_number]
```

4.7 Symbolic Offset Access into Options and Statistics Arrays

Access to the options and statistics arrays used with the callable Convert and Convert/Reclaim utilities has been facilitated by defining the array elements as named longwords in a structure and by providing symbolic names for the element numbers. The $CONVDEF module in the STARLET macro library has been expanded to provide these symbolic definitions.

4.7.1 Access to Qualifier Options Array

Programmers using the callable Convert routine CONV$PASS_OPTIONS can now specify qualifiers to the CONVERT command using zero-based, symbolic offsets (CONV$K_...) into the variable-length array of longwords which lists the qualifier options. The array is now also defined as a structure (CONV$OPTIONS) of named longwords (CONV$L_...) to support access by high-level programming languages.

Table 4–3 lists the array elements by number and by symbol and includes the default values for each qualifier option. Note that the first element contains the number of valid elements in the array.

Table 4–3 CONVERT Options Array

Array Element	Symbolic Value	Default Value (in Longwords)	Default CONVERT Command Value
0	CONV$L_OPTIONS_COUNT	No default	N/A
1	CONV$L_CREATE	1	/CREATE
2	CONV$L_SHARE	0	/NOSHARE
3	CONV$L_FAST	1	/FAST_LOAD

(continued on next page)

Table 4–3 (Cont.) CONVERT Options Array

Array Element	Symbolic Value	Default Value (in Longwords)	Default CONVERT Command Value
4	CONV$L_MERGE	0	/NOMERGE
5	CONV$L_APPEND	0	/NOAPPEND
6	CONV$L_SORT	1	/SORT
7	CONV$L_WORK_FILES	2	/WORK_FILES=2
8	CONV$L_KEY	0	/KEY=0
9	CONV$L_PAD	0	/NOPAD
10	CONV$L_PAD_CHARACTER	0	Pad character=0
11	CONV$L_TRUNCATE	0	/NOTRUNCATE
12	CONV$L_EXIT	0	/NOEXIT
13	CONV$L_FIXED_CONTROL	0	/NOFIXED_ CONTROL
14	CONV$L_FILL_BUCKETS	0	/NOFILL_BUCKETS
15	CONV$L_READ_CHECK	0	/NOREAD_CHECK
16	CONV$L_WRITE_CHECK	0	/NOWRITE_CHECK
17	CONV$L_FDL	0	/NOFDL
18	CONV$L_EXCEPTION	0	/NOEXCEPTION
19	CONV$L_PROLOG	No default	System or process default

4.7.2 Access to Conversion Statistics Array

Programmers using the callable Convert routine CONV$CONVERT can now request conversion statistics using zero-based, symbolic offsets (CONV$K_...) into the variable-length array of longwords which contains the statistics. The array is now also defined as a structure (CONV$STATISTICS) of named longwords (CONV$L_...) to support access by high-level programming languages.

Table 4–4 lists the array elements by number and by symbol. The first element is used to specify the number of statistics to return by array order. For example, if you assign the symbol CONV$L_STATISTICS_COUNT the value 2, the routine returns the statistics from the first two statistics elements:

- Number of files converted
- Number of records converted

Table 4–4 Conversion Statistics Array

Array Element	Field Name	Description
0	CONV$L_STATISTICS_COUNT	Number of statistics specified
1	CONV$L_FILE_COUNT	Number of files

(continued on next page)

Table 4–4 (Cont.) Conversion Statistics Array

Array Element	Field Name	Description
2	CONV$L_RECORDS_COUNT	Number of records
3	CONV$L_EXCEPTION_RECORDS_COUNT	Number of exception records
4	CONV$L_VALID_RECORDS_COUNT	Number of valid records

4.7.3 Access to Reclamation Statistics Array

Programmers using the callable Convert/Reclaim routine CONV$RECLAIM can now request reclamation statistics using zero-based, symbolic offsets (RECL$K_...) into the variable-length array of longwords which contains the statistics. The array is now also defined as a structure (RECL$STATISTICS) of named longwords (RECL$L_...) to support access by high-level programming languages.

Table 4–5 lists the array elements by number and by symbol. The first element is used to specify one or more statistics by array order. For example, if you assign the symbol RECL$L_STATISTICS_COUNT the value 3, the routine returns the statistics from the first three statistics elements:

- Data buckets scanned
- Data buckets reclaimed
- Index buckets reclaimed

Table 4–5 Bucket Reclamation Statistics Array

Array Element	Field Name	Description
0	RECL$L_STATISTICS_COUNT	Number of statistics specified
1	RECL$L_SCAN_COUNT	Data buckets scanned
2	RECL$L_DATA_COUNT	Data buckets reclaimed
3	RECL$L_INDEX_COUNT	Index buckets reclaimed
4	RECL$L_TOTAL_COUNT	Total buckets reclaimed

For more detailed information, see the *OpenVMS Utility Routines Manual*.

4.8 New Optional Argument Supported for Calls to FDL$CREATE

The FDL$CREATE routine now accepts the name of the default FDL file or the default FDL specification as an optional routine call argument. The optional **dflt_fdl_spc** argument is the address of a character string descriptor pointing to either the default FDL file or the default FDL specification. If the FDL$V_DEFAULT_STRING flag is set in the **flags** argument, FDL$CREATE interprets this argument as an FDL specification in string form. Otherwise, FDL$CREATE interprets this argument as an FDL file name.

With this argument, you can specify default FDL attributes. FDL$CREATE processes the attributes specified in the character string descriptor, unless you override these attributes with the attributes you specify in the **fdl_spec** argument.

You can also code the FDL defaults directly into your program, typically with an FDL specification in string form.

This argument is optional.

For more information, see the *OpenVMS Utility Routines Manual*.

4.9 Programming Support for File Shelving

OpenVMS AXP Version 6.1 supports the layered product POLYCENTER HSM for OpenVMS. This is a file shelving facility that lowers the average cost of storing data that is rarely used.

There are two new file bits stored in the file header. One tells you whether the file is shelved, the other tells you whether the file is shelvable. The following table shows you how to access these bits using RMS and the $QIO system service:

To...	Using RMS	Using $QIO
Read the shelved bit	Use the XAB$_UCHAR_SHELVED item code. Its values include: 1 = Shelved 0 = Online	Use the FH2$V_SHELVED bit of the ATR$C_UCHAR attribute. Its values include: 1 = Shelved 0 = Online
Read the shelvable bit	Use the XAB$_UCHAR_NOSHELVABLE item code. Its values include: 1 = Not shelvable 0 = Shelvable	Use the FH2$V_NOSHELVABLE bit of the ATR$C_UCHAR attribute. Its values include: 1 = Not shelvable 0 = Shelvable
Write the shelvable bit	Use the XAB$_UCHAR_NOSHELVABLE item code. Its values include: 1 = Not shelvable 0 = Shelvable	Use the FH2$V_NOSHELVABLE bit of the ATR$C_UCHAR attribute. Its values include: 1 = Not shelvable 0 = Shelvable

You cannot write the shelved bit.

There is a new bit in the process control block (PCB) that controls whether automatic unshelving is turned on. The following table shows you how to read and write this bit.

To...	Use...
Read the automatic unshelving bit	$GETJPI. Use the PCB$V_NOUNSHELVE bit of the JPI$_STS2 item code. Its values include: 1 = No automatic unshelving 0 = Automatic unshelving

To...	Use...
Write the automatic unshelving bit	The new $SETSHLV system service. See the *OpenVMS System Services Reference Manual: GETQUI–Z*.

4.10 LOGINOUT Callout Routines

The LOGINOUT callout routines enable user installations to merge site-specific login policies and events with OpenVMS LOGINOUT policies and events.

With this release, the LOGINOUT callout routines have been added to user documentation. For details, see the *OpenVMS Utility Routines Manual*.

4.11 Linker Enhancement

The following sections describe new AXP enhancements to the OpenVMS Linker utility. Refer to the online version of the *OpenVMS Linker Utility Manual* for additional information on these enhancements.

4.11.1 New /DSF Qualifier

The /DSF qualifier directs the linker to create a file called a debug symbol file (DSF) for use by the OpenVMS AXP System-Code Debugger. The default is /NODSF. The /DSF qualifier can be used with the /NOTRACEBACK qualifier to suppress the appearance of SYS$IMGSTA in the image's transfer array. The /DSF qualifier has no effect on the contents of the image, including the image header.

The /DSF and /DEBUG qualifiers are not mutually exclusive. However, the combination is not generally useful. The debug bit in the image header will be set and SYS$IMGSTA will be included in the transfer array, but there will be no information for the symbolic debugger included in the image. The DSF file will be generated as usual.

Use the following format:

```
LINK/DSF[=file-spec]
```

See *OpenVMS AXP Device Support: Developer's Guide* for guidelines on using the OpenVMS AXP System-Code Debugger.

4.11.2 New /ATTRIBUTES Qualifier for COLLECT= Option

The COLLECT= option directs the linker to place the specified program section (or program sections) into the specified cluster. A new qualifier /ATTRIBUTES directs the linker to mark the cluster 'cluster-name' with the indicated qualifier keyword value. This qualifier is used to build AXP drivers.

Use the following format:

```
COLLECT=cluster-name [/ATTRIBUTES={RESIDENT | INITIALIZATION_CODE}],-
  psect-name[,...])
```

Qualifier Values

RESIDENT
Marks the cluster 'cluster-name' as RESIDENT so that the image section created from that cluster has the EISD$V_RESIDENT flag set. This will cause the loader to map the image section into nonpaged memory.

INITIALIZATION_CODE
Marks the cluster 'cluster-name' as INITIALIZATION_CODE so that the image section created from that cluster has the EISD$V_INITALCOD flag set. The initialization code will be executed by the loader. This keyword is specifically intended for use with program sections from modules SYS$DOINIT and SYS$DRIVER_INIT in STARLET.OLB.

See *OpenVMS AXP Device Support: Developer's Guide* for guidelines for using this qualifier.

4.12 System Dump Analyzer for AXP

The following features are new for the OpenVMS AXP System Dump Analyzer (SDA) utility. For more information, please refer to *OpenVMS AXP System Dump Analyzer Utility Manual*.

4.12.1 Using SDA CLUE Commands to Obtain and Analyze Crash Dump Information

On OpenVMS AXP systems, the Crash Log Utility Extractor (CLUE) extension commands can summarize information provided by certain standard SDA commands and also provide additional detail for some SDA commands. These SDA CLUE commands can interpret the contents of the dump to perform additional analysis.

4.12.1.1 Understanding SDA CLUE

When rebooting after a system failure, SDA is automatically invoked and SDA CLUE commands analyze and save summary information from the crash dump file in CLUE history and listing files.

The CLUE HISTORY command updates the history file pointed to by the logical name CLUE$HISTORY with a one line entry and the major crash dump summary information.

The CLUE listing file contains more detailed information about the system crash and is created in the directory pointed to by the logical name CLUE$COLLECT. The following information is included in the CLUE listing file:

- Crash dump summary
- System configuration
- Stack decoder
- Page and swap files
- Memory management statistics
- Process DCL recall buffer
- Active XQP processes
- XQP cache header

The CLUE list file can be printed immediately or saved for later examination.

SDA is invoked and executes the specified commands only when the system boots immediately after a system failure. If the system is booting for any other *reason*, SDA exits.

4.12.1.2 Displaying Data Using SDA CLUE Commands

Invoke CLUE commands at the SDA prompt as follows:

```
SDA> CLUE CRASH
```

All CLUE commands can be used when analyzing crash dumps; the only CLUE commands that are not allowed when analyzing a running system are CLUE CRASH, CLUE ERRLOG, CLUE HISTORY and CLUE STACK.

4.12.2 New Qualifier for SET PROCESS and SHOW PROCESS Commands

In order to display information about a process, you can use the new qualifier /ADDRESS=pcb-address with the SET PROCESS or SHOW PROCESS command to specify the process control block (PCB) address of a process.

4.13 Digital Portable Mathematics Library (DPML)

The Digital Portable Mathematics Library (DPML) now provides 128-bit IEEE extended-precision number support for the X_FLOAT and X_FLOAT_COMPLEX data types. In addition, Table 4–6 lists the new mathematics routines provided for this release.

Table 4–6 New DPML Routines

Name	Function
acosh	Hyperbolic arc cosine
asinh	Hyperbolic arc sine
j0	Bessel function
j1	Bessel function
jn	Bessel function
cabs	Complex absolute value
cbrt	Cube root
copysign	Copy sign
drem	Remainder
erf	Error function
erfc	Complementary error function
expm1	Exponential function
fabs	Absolute value
finite	Check for finite value
fmod	Modulo remainder
fp_class	Classifies IEEE floating-point values
frexp	Convert to fraction and integral power of 2
isnan	Check for NaN value
ldexp	Multiply by an integral power of 2

(continued on next page)

Table 4–6 (Cont.) New DPML Routines

Name	Function
lgamma	Computes the logarithm of the gamma function
log1p	Logarithm function
logb	Convert to double-precision floating-point
modf	Split a floating-point number into a fractional and integer part
nextafter	Next machine number after
nint	Round to the nearest integer
remainder	Remainder
rint	Returns the nearest integral value
scalb	Exponent adjustment
unordered	Checks for x unordered with respect to y
y0	Bessel function
y1	Bessel function
yn	Bessel function

For specific data-type entry-points for each routine, see the *Digital Portable Mathematics Library* manual.

4.14 X_Floating Atomic Data Types for Calling Routines

OpenVMS on AXP systems now supports the IEEE X_floating atomic data types for passing arguments when calling routines in high-level language environments. Table 4–7 lists the new atomic data types with descriptor symbols, codes, and descriptions.

Table 4–7 X_Floating Atomic Data Types

Symbol	Code	Name/Description
DSC$K_DTYPE_FX	57	X_floating 128-bit IEEE X_floating-point quantity representing an extended-precision number.
DSC$K_DTYPE_FXC	58	X_floating complex Ordered pair of X_floating-point quantities representing an extended-precision complex number. The lower addressed quantity is the real part; the higher addressed quantity is the imaginary part.

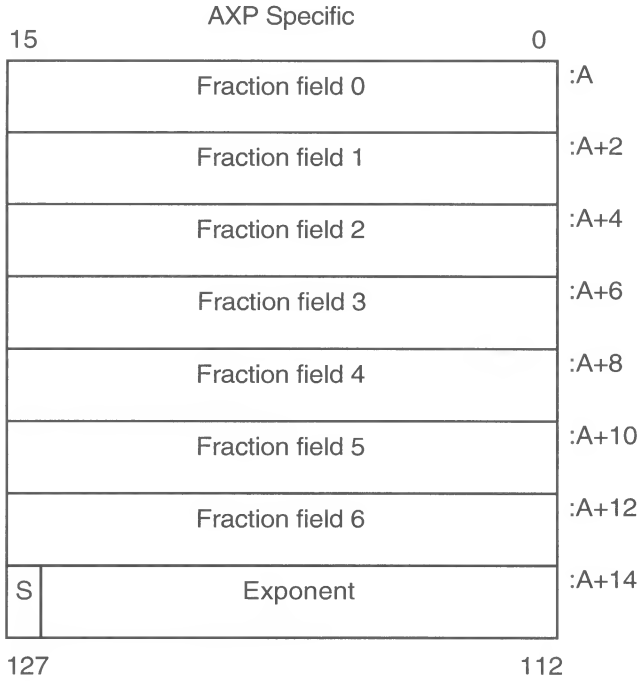
For more information about IEEE atomic data types, see the *OpenVMS Calling Standard*.

4.14.1 X_Floating Data Type and Datum

OpenVMS AXP systems now support an X_floating data type in the standard set of **floating_point** (S, T, and X) data types. The structure of the 128-bit X_floating data type datum is shown in Figure 4–1. An X_floating datum occupies 16 contiguous bytes in memory or two consecutive floating-point registers. It contains seven fraction fields (0–6). Note that fraction field 6 holds the most significant bits and the field 0 extension holds the least significant portion of the fractional number.

This section expands the IEEE S_ and T_floating-point support reference for OpenVMS data type usage described in the *OpenVMS Programming Interfaces: Calling a System Routine* to include the X_floating-point types.

Figure 4–1 Structure of an X_Floating Datum

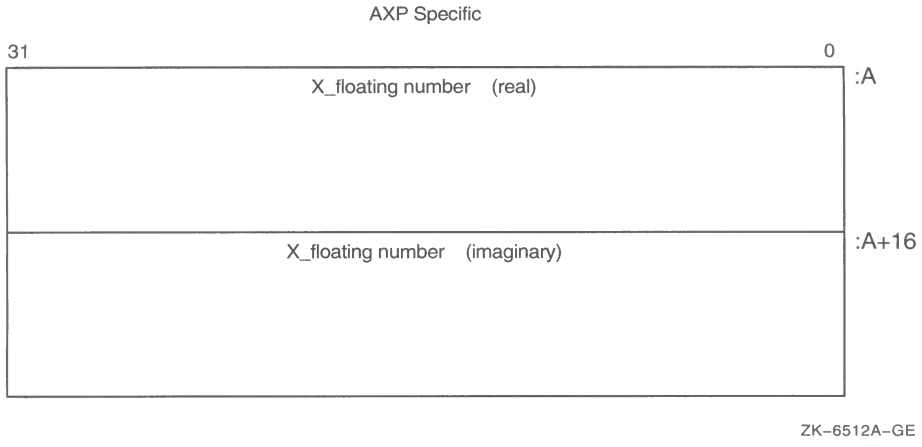


4.14.2 X_Floating Complex Data Type and Structure

OpenVMS AXP systems now support an X_floating complex data type in the standard set of floating-point **complex_number** OpenVMS usage data types offered. The overall structure of the floating-point complex_number data type is shown in Figure 4–2. An X_floating complex_number (real, imaginary) is composed of two X_floating-point numbers: the first is the real part of the complex_number; the second is the imaginary part.

This section expands the IEEE S_ and T_floating-point complex_number support reference for OpenVMS data type usage described in the *OpenVMS Programming Interfaces: Calling a System Routine* to include the X_floating-point complex_number types.

Figure 4–2 Structure of an X_Floating Complex_Number Datum



4.15 C++ Data Type Declarations

OpenVMS on AXP systems now provides data type declarations for passing arguments when calling routines in the high-level C++ language. Section 4.15.1 describes both C and C++ data type implementations. For more information about argument-passing data types, see the *OpenVMS Programming Interfaces: Calling a System Routine*.

4.15.1 C and C++ Implementations

Table 4–8 lists the OpenVMS data types and their corresponding C and C++ data type declarations.

Table 4–8 C and C++ Implementations

OpenVMS Data Types	C and C++ Declarations
access_bit_names	User defined[1]
access_mode	unsigned char
address	User defined[1] *pointer[2,4]
address_range	int *array [2] [2,3,4]
arg_list	User defined[1]
ast_procedure	Pointer to function[2]
boolean	unsigned long int
byte_signed	char
byte_unsigned	unsigned char
channel	unsigned short int
char_string	char array[*n*][3,5]
complex_number	User defined[1]
cond_value	unsigned long int
context	unsigned long int
date_time	User defined[1]
device_name	char array[*n*][3,5]
ef_cluster_name	char array[*n*][3,5]
ef_number	unsigned long int
exit_handler_block	User defined[1]
fab	#include <fab.h> struct FAB
file_protection	unsigned short int or user defined[1]
floating_point	float, double, or long double
function_code	unsigned long int or user defined[1]
identifier	unsigned long int *pointer[2,4]
‡invo_context_blk	#include <libicb.h> struct invo_context_blk
‡invo_handle	unsigned long int
io_status_block	User defined[1]
item_list_2	User defined[1]
item_list_3	User defined[1]
item_list_pair	User defined[1]
item_quota_list	User defined[1]

[1]The declaration of a user-defined data structure depends on how the data will be used. Such data structures can be declared in a variety of ways, each of which is suitable only to specific applications.

[2]C and C++ pointers are declared with special syntax and are associated with the data type of the object being pointed to. This object is often user defined.

[3]The term array denotes the syntax of a C or C++ array declaration.

[4]The data type specified can be changed to any valid C or C++ data type.

[5]The size of the array must be substituted for *n*.

‡AXP specific.

(continued on next page)

Table 4–8 (Cont.) C and C++ Implementations

OpenVMS Data Types	C and C++ Declarations
lock_id	unsigned long int
lock_status_block	User defined[1]
lock_value_block	User defined[1]
logical_name	char array[*n*][3,5]
longword_signed	long int
longword_unsigned	unsigned long int
mask_byte	unsigned char
mask_longword	unsigned long int
mask_quadword	User defined[1]
mask_word	unsigned short int
mechanism_args	#include <chfdef.h> struct chf$mech_array
null_arg	unsigned long int
octaword_signed	User defined[1]
octaword_unsigned	User defined[1]
page_protection	unsigned long int
procedure	Pointer to function[2]
process_id	unsigned long int
process_name	char array[*n*][3,5]
quadword_signed	User defined[1]
quadword_unsigned	User defined[1]
rights_holder	User defined[1]
rights_id	unsigned long int
rab	#include <rab.h> struct RAB
section_id	User defined[1]
section_name	char array[*n*][3,5]
system_access_id	User defined[1]
time_name	char array[*n*][3,5]
transaction_id	User defined[1]
uic	unsigned long int
user_arg	User defined[1]
varying_arg	User defined[1]
vector_byte_signed	char array[*n*][3,5]
vector_byte_unsigned	unsigned char array[*n*][3,5]

[1]The declaration of a user-defined data structure depends on how the data will be used. Such data structures can be declared in a variety of ways, each of which is suitable only to specific applications.

[2]C and C++ pointers are declared with special syntax and are associated with the data type of the object being pointed to. This object is often user defined.

[3]The term array denotes the syntax of a C or C++ array declaration.

[5]The size of the array must be substituted for *n*.

(continued on next page)

Table 4–8 (Cont.) C and C++ Implementations

OpenVMS Data Types	C and C++ Declarations
vector_longword_signed	long int array[*n*][3,5]
vector_longword_unsigned	unsigned long int array[*n*][3,5]
vector_quadword_signed	User defined[1]
vector_quadword_unsigned	User defined[1]
vector_word_signed	short int array[*n*][3,5]
vector_word_unsigned	unsigned short int array[*n*][3,5]
word_signed	short int
word_unsigned	unsigned short int

[1]The declaration of a user-defined data structure depends on how the data will be used. Such data structures can be declared in a variety of ways, each of which is suitable only to specific applications.

[3]The term array denotes the syntax of a C or C++ array declaration.

[5]The size of the array must be substituted for *n*.

4.16 New Run-Time Library Support Available

The common OpenVMS Version 6.1 Run-Time Library environment now supports the following high level languages:

- Ada
- BASIC
- C++
- Pascal
- PL/I

For more information on the Run-Time Library, see the *OpenVMS Programming Interfaces: Calling a System Routine* and the specific compiler documentation.

4.17 Native MACRO-32 Compiler

As of this release, the MACRO–32 Compiler for OpenVMS AXP is a native compiler. The earlier versions were translated. Most users of earlier versions will notice significant improvements in compilation times when using this native compiler.

Note

The MACRO–32 compiler is provided for porting VAX MACRO code to OpenVMS AXP systems. For any new development, Digital recommends the use of mid- and high-level languages.

For more information about the MACRO–32 compiler, see *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code* and the *OpenVMS AXP Version 6.1 Release Notes*.

4.18 System Services Features

This section contains information about new features for system services.

4.18.1 Batch and Print Queuing System Services

The following sections describe new features for the system services supporting the batch and print queuing system.

For detailed information, see the *OpenVMS System Services Reference Manual.*

4.18.1.1 New Features for $GETQUI

OpenVMS AXP Version 6.1 allows multiple streams of operation for the $GETQUI system service. This support allows simultaneous requests of any valid wildcard function. A new argument, **context**, replaces the **nullarg** argument to $GETQUI to support this feature.

In addition, the following codes were added on OpenVMS AXP Version 6.1:

- New function code, QUI$_DISPLAY_MANAGER
- Four new item codes valid for the new QUI$_DISPLAY_MANAGER function code:
  - QUI$_MANAGER_NAME
  - QUI$_MANAGER_NODES
  - QUI$_MANAGER_STATUS
  - QUI$_QUEUE_DIRECTORY
- Modified item code, QUI$_SCSNODE_NAME, valid for the new QUI$_DISPLAY_MANAGER function code
- New bit, QUI$V_INACCESSIBLE, valid for the QUI$_QUEUE_FLAGS item code

4.18.1.2 New Features for $SNDJBC

The following new features are available with the $SNDJBC system service:

- Function code, SJC$_DELETE_QUEUE_MANAGER
- Two new item codes:
  - SJC$_ADD_QUEUE_MANAGER, valid for the SJC$START_QUEUE_MANAGER function code
  - SJC$_QUEUE_MANAGER_NAME, valid for the following function codes:

    SJC$_CREATE_QUEUE
    SJC$_DELETE_QUEUE_MANAGER
    SJC$_DISABLE_AUTOSTART
    SJC$_ENABLE_AUTOSTART
    SJC$_START_QUEUE_MANAGER
    SJC$_STOP_ALL_QUEUES_ON_NODE
    SJC$_STOP_QUEUE_MANAGER

4.18.2 Cluster Configuration Event Notification System Services

Three new system services have been added on OpenVMS AXP Version 6.1 to support cluster configuration event notification. Table 4–9 describes these services.

Table 4–9 New Cluster Configuration Event Notification Services

Service Name	Description
$CLRCLUEVT	Removes one or more notification requests previously established by a call to SYS$SETCLUEVT.
$SETCLUEVT	Establishes a request for notification when a cluster configuration event occurs.
$TSTCLUEVT	Simulates the occurrence of a cluster configuration event to test the functionality of the notification asynchronous system trap.

For detailed information, see the *OpenVMS System Services Reference Manual.*

4.18.3 DECdtm System Services Enhancements

Several new condition values are now returned by the DECdtm system services on OpenVMS AXP Version 6.1.

The $ABORT_TRANS system service now also returns the following condition values, in addition to the values that it returned previously:

- SS$_BADREASON
- SS$_CURTIDCHANGE
- SS$_NOTORIGIN
- SS$_TPDISABLED

The $END_TRANS system service now also returns the following condition values, in addition to the values that it returned previously:

- SS$_CURTIDCHANGE
- SS$_NOLOG
- SS$_NOTORIGIN
- SS$_TPDISABLED

The $START_TRANS system service now also returns the SS$_CURTIDCHANGE condition value, in addition to the values that it returned previously.

For a full description of these system services, see the *OpenVMS System Services Reference Manual.*

4.18.4 $FILESCAN System Service Enhancements

Several enhancements have been made to the $FILESCAN system service with OpenVMS AXP Version 6.1 to support the processing of full names.

Two new arguments, **auxout** and **retlen**, have been added.

The **auxout** argument can be used to specify an auxiliary output buffer into which the $FILESCAN system service copies an entire source string. When you specify the **auxout** argument, quotation information is reduced and simplified for the primary node only.

The **retlen** argument is the address of a word into which the $FILESCAN system service writes the length of the auxiliary buffer string.

Three new flag bits have been added to correspond with three new file specifications. Table 4–10 shows the file specification component that corresponds to the symbolic name of each flag bit.

Table 4–10 New $FILESCAN Flag Bits

Symbolic Name	Corresponding Component
FSCN$V_NODE_PRIMARY	Primary (first) node name
FSCN$V_NODE_ACS	Access control string of primary node
FSCN$V_NODE_ SECONDARY	Secondary (additional) node information

In addition, three new item codes have been added. Table 4–11 describes the new item codes.

Table 4–11 New $FILESCAN Item Codes

Item Code	Description
FSCN$_NODE_PRIMARY	Retrieves the primary node specification.
FSCN$_NODE_ACS	Retrieves the access control string.
FSCN$_NODE_ SECONDARY	Retrieves the secondary node specification.

For detailed information, see the *OpenVMS System Services Reference Manual.*

4.18.5 $GET_ARITH_EXCEPTION System Service

The $GET_ARITH_EXCEPTION system service returns information about the exception context for a given arithmetic exception.

For detailed information, see the *OpenVMS System Services Reference Manual.*

4.18.6 $GETDVI System Service Enhancements

Several new item codes have been added on OpenVMS AXP Version 6.1 for terminal devices.

Table 4–12 describes the new item codes.

Table 4–12 New $GETDVI Item Codes

Item Code	Description
DVI$_TT_CHARSET	Returns the character sets supported by the terminal.
DVI$_TT_CS_HANGUL	Indicates whether or not the device supports the DEC Hangul coded character set.
DVI$_TT_CS_HANYU	Indicates whether or not the device supports the DEC Hanyu coded character set.

(continued on next page)

Table 4–12 (Cont.) New $GETDVI Item Codes

Item Code	Description
DVI$_TT_CS_HANZI	Indicates whether or not the device supports the DEC Hanzi coded character set.
DVI$_TT_CS_KANA	Indicates whether or not the device supports the DEC Kana coded character set.
DVI$_TT_CS_KANJI	Indicates whether or not the device supports the DEC Kanji coded character set.
DVI$_TT_CS_THAI	Indicates whether or not the device supports the DEC Thai coded character set.

A DVI$_DEVICE_TYPE_NAME item code has been added on OpenVMS AXP Version 6.1. When you specify DVI$_DEVICE_TYPE_NAME, the $GETDVI system service returns a string identifying the type of the device about which information was requested.

For detailed information, see the *OpenVMS System Services Reference Manual*.

4.18.7 New Argument for $GETUAI and $SETUAI System Services

A **contxt** argument has been added to the $GETUAI and $SETUAI system services. The argument is used to maintain user authorization file context.

For detailed information, see the *OpenVMS System Services Reference Manual*.

4.18.8 $IEEE_SET_FP_CONTROL System Service

The $IEEE_SET_FP_CONTROL system service modifies the software IEEE floating-point control register and, optionally, returns the previous register value.

For detailed information, see the *OpenVMS System Services Reference Manual*.

4.18.9 $SCHED System Service

The $SCHED system service affects process scheduling on an OpenVMS system. This service is intended for use by class scheduler processes.

The $SCHED system service allows users to write class schedulers. A sample class scheduler, called CLASS.C, is located in SYS$EXAMPLES. This sample class scheduler has the following uses:

- It can, with minor site-specific modifications, be used as a class scheduler.
- It shows the call formats for writing a class scheduler.

For detailed information, see the *OpenVMS System Services Reference Manual*.

4.18.10 Security System Services

Table 4–13 contains a summary of new security system services.

Table 4–13 Summary of New Features for Security System Services

Service	Description
$AUDIT_EVENT	Appends an event message to the system security audit log file or sends an alarm to a security operator terminal.
$CHECK_PRIVILEGE	Determines whether the caller has the specified privileges or identifier.
$CREATE_USER_PROFILE	Returns a security profile for the specified user.
$GET_SECURITY	Returns information about security elements of a selected object.
$SET_RESOURCE_DOMAIN	Controls the association between a calling process and resource domains.
$SET_SECURITY	Modifies the security elements of a protected object.
$SUBSYSTEM	Saves or restores the process rights for the current protected subsystem.

For detailed information, see Section 3.18 and the *OpenVMS System Services Reference Manual*.

4.18.11 Time Conversion Service

OpenVMS AXP Version 6.1 system time is based on the local time effective in the time zone where the system is installed. For example, systems simultaneously installed on the East Coast and West Coast of the United States have system times that differ by three hours. If a distributed database application is running on the two systems, it is possible for an older entry on the East Coast system to supersede a newer entry on the West Coast system. OpenVMS AXP Version 6.1 provides a time conversion service to overcome this problem.

The time conversion service converts the local system time to Coordinated Universal Time (UTC), which has largely replaced Greenwich Mean Time (GMT) as a timekeeping standard. After you configure the time conversion service, UTC time is available to applications through the $GETUTC system service. The $GETUTC service enables applications to obtain timestamps based on a single, widely accepted time reference, regardless of where the applications are running. It also returns the current time in 128-bit UTC format.

To use the time conversion service, you must configure it by setting your system's Time Differential Factor (TDF), which is the number of minutes that the local system time differs from UTC. The conversion service applies the TDF to the local system time to calculate the UTC.

Table 4–14 describes the new timer system services.

Table 4–14 New Timer System Services

Service	Description
$ASCUTC	Converts an absolute time from 128-bit UTC format to an ASCII string.
$BINUTC	Converts an ASCII string to an absolute time value in the 128-bit UTC format.

(continued on next page)

Table 4–14 (Cont.) New Timer System Services

Service	Description
$GETUTC	Enables applications to obtain timestamps based on a single, widely accepted time reference, regardless of where the applications are running. Returns the current time in 128-bit UTC format.
$NUMUTC	Converts absolute time in 128-bit UTC format into its numeric components. The numeric components are returned in local time.
$TIMCON	Converts 128-bit UTC format to 64-bit system format or 64-bit system format to 128-bit UTC format based on the value of the convert flag.

For detailed information, see the *OpenVMS System Services Reference Manual*.

4.19 Support for the Movefile Subfunction

OpenVMS AXP Version 6.1 now supports the movefile subfunction, as documented for OpenVMS VAX Version 6.0. This subfunction allows you to move the contents of a file, or part of the contents of a file, to another disk location.

For more detailed information, see the *OpenVMS I/O User's Reference Manual*.

4.20 New File Information Block (FIB) Changes

As of Version 6.1 of OpenVMS, fields, subfields, subfunctions, and major functions of the file information block (FIB) are supported on both OpenVMS VAX systems and OpenVMS AXP systems. For more information, see the *OpenVMS I/O User's Reference Manual*.

4.20.1 FIB$L_ACCTL (Access Control) Subfunctions

Table 4–15 contains the subfields for FIB$L_ACCTL access control subfunctions that are no longer OpenVMS VAX system specific.

Table 4–15 FIB Access Control Fields

Field	Subfields	Meaning
FIB$L_ACCTL		Specifies field values that control access to the file. The following access control bits in the subfields are applicable to the access subfunction.
	FIB$V_CONTROL	Set for control access. If this bit is set, you cannot access the file if you do not have control access.
	FIB$V_NO_READ_DATA	Set to deny read access to the file.

4.20.2 IO$_CREATE (Create File) Major Functions

Table 4–16 contains the field and subfields of IO$_CREATE create file function code that are no longer OpenVMS VAX system specific.

Table 4–16 IO$_CREATE and the File Information Block

Field	Subfields	Meaning
FIB$L_STATUS		Access status. Programmers can control the security information being propagated as well as the source of this information by setting the following bits in the subfields.
	FIB$V_DIRACL	Propagate the ACL from the parent directory to the file, assuming the file is a directory file.
	FIB$V_EXCLPREVIOUS	Set to indicate that propagation may not occur from a previous version of the file.
	FIB$V_ALIAS_ENTRY	Set on any file system operation where the directory backlink in the file header is different (and nonzero) from the directory id specified in the FIB.
	FIB$V_NOCOPYACL	Set to indicate that the ACL should not be propagated from the parent directory (or a previous version of the file) to the file.
	FIB$V_NOCOPYOWNER	Set to indicate that the owner UIC should not be propagated from the parent directory (or a previous version of the file) to the file.
	FIB$V_NOCOPYPROT	Set to indicate that the UIC-based protection should not be propagated from the parent directory (or a previous version of the file) to the file.
	FIB$V_PROPAGATE	Propagate attributes from the parent directory (or previous version of the file). If you set the FIB$V_NOCOPYACL, FIB$V_NOCOPYOWNER, or FIB$V_NOCOPYPROT bits, you must also set FIB$V_PROPAGATE or a SS$_BADPARAM error results.

4.20.3 IO$_MODIFY (Modify File) Major Functions

Table 4–17 contains the field and subfields of IO$_MODIFY modify file function code that are no longer OpenVMS VAX system specific.

Table 4–17 IO$_MODIFY and the File Information Block

Field	Subfields	Meaning
FIB$L_STATUS		Access status. Applies to all major functions. Programmers can control the security information being propagated as well as the source of this information by setting the following bits in the subfields.
	FIB$V_DIRACL	Propagate the ACL from the parent directory to the file, assuming the file is a directory file.
	FIB$V_EXCLPREVIOUS	Set to indicate that propagation may not occur from a previous version of the file.
	FIB$V_ALIAS_ENTRY	Set on any file system operation where the directory backlink in the file header is different (and nonzero) from the directory id specified in the FIB.

(continued on next page)

Table 4–17 (Cont.) IO$_MODIFY and the File Information Block

Field	Subfields	Meaning
	FIB$V_NOCOPYACL	Set to indicate that the ACL should not be propagated from the parent directory (or a previous version of the file) to the file.
	FIB$V_NOCOPYOWNER	Set to indicate that the owner UIC should not be propagated from the parent directory (or a previous version of the file) to the file.
	FIB$V_NOCOPYPROT	Set to indicate that the UIC-based protection should not be propagated from the parent directory (or a previous version of the file) to the file.
	FIB$V_PROPAGATE	Propagate attributes from the parent directory (or previous version of the file). If you set the FIB$V_NOCOPYACL, FIB$V_NOCOPYOWNER, or FIB$V_NOCOPYPROT bits, you must also set FIB$V_PROPAGATE or a SS$_BADPARAM error results.

4.21 LAT Symbiont Customization

In previous versions of the OpenVMS operating system, you could modify a print symbiont but not a LAT symbiont. With OpenVMS AXP Version 6.1, you can use the LAT protocol in a modified symbiont. This change has no effect on system management tasks such as setting up, creating, and managing queues.

The programming interface to PSM$PRINT has been enhanced to include the ability to use the newly added LAT protocol. The following section describes the two new arguments.

Format

PSM$PRINT [streams] [,bufsiz] [,worksiz] [,maxqios] [,options]

Arguments

maxqios

OpenVMS usage: longword_unsigned
type: longword (unsigned)
access: read only
mechanism: by reference

Specifies the maximum number of outstanding $QIOs that a LAT symbiont stream will generate. The symbiont process quotas for the symbiont process must be large enough to handle the maximum number of QIOs, times the number of streams should all streams become stalled. Use a number between 2 and 32. Ten is suggested for normal printing capabilities; a larger number for high-speed printers.

options

OpenVMS usage: longword_unsigned
type: longword (unsigned)
access: read only
mechanism: by reference

Longword bit vector that specifies symbiont options. The following is the valid option flag:

Option	Description
PSM$V_LAT_PROTOCOL	Specifies that the LAT protocol is used

Note that using the LAT_PROTOCOL option carries the following restrictions:

- Replacement of the output and job completion routines will be overridden.
- The output device must be a LAT device.

4.22 LAT Programming Features

This section contains information about new LAT programming features.

4.22.1 LAT $QIO Function Modifier

With OpenVMS AXP Version 6.1, $QIO has a new LAT function modifier: IO$M_LT_QUE_CHG_NOTIF. The IO$M_LT_QUE_CHG_NOTIF function allows a process to enable an attention asynchronous system trap (AST) and is used with the LAT $QIO connect request. The IO$M_LT_QUE_CHG_NOTIF function is available only for APPLICATION and FORWARD LAT devices.

If a $QIO connect request has been issued to a remote node and that request has been queued, the attention AST will be set each time the queue position changes. This AST can be used as long as the $QIO connect request is queued. Like a CONTROL-Y AST, it is set only once; it must be reenabled after each completion.

If the LAT $QIO connect succeeds or if a LAT connection exists for the intended service, the AST completes with the SS$_DEVACTIVE status code.

If the LAT device does not have the queued characteristic, issuing the IO$M_LT_QUE_CHG_NOTIF function results in the return of SS$_DEVREQERR status code.

The implementation of IO$M_LT_QUE_CHG_NOTIF is shown in the following C example:

```
status = sys$qiow  (
                0,                  /* efn                */
                ltchannel,          /* channel            */
                IO$_TTY_PORT|IO$M_LT_QUE_CHG_NOTIF,
                                    /* function           */
                q_iosb,             /* iosb               */
                0,                  /* astadr             */
                0,                  /* astprm             */
                queue_pos_change,   /* P1 = ast routine   */
                0, 0, 0, 0, 0);     /* P2 through P6 not used */
```

When a queue position change occurs, the AST routine is called with a 32-bit value. If this value is 0, then the LAT connect $QIO is about to complete, if it has not already. If the value is not 0, the lower word of 16-bits indicates the service queue position, and the upper word of 16-bits indicates the node queue position.

For more detailed information, see the *OpenVMS I/O User's Reference Manual.*

4.22.2 LAT SENSEMODE Argument

The LAT SENSEMODE $QIO function accepts a new argument for the P3 argument. The new argument for P3 is **Queue Entry** (LAT$C_ENT_QUEUE_ENTRY). The **Queue Entry** argument indicates queue entry entities.

For more detailed information, see *OpenVMS I/O User's Reference Manual.*

4.22.3 LAT SENSEMODE Item Codes

This section describes the item codes for port node counters, node counters, protocol errors, service counters subblocks, port entities, and queue entries.

4.22.3.1 LAT SENSEMODE Item Codes for Port Node Counters Item Codes

On OpenVMS AXP systems, port counters information is presented as a counters subblock. The subblock item code LAT$_ITM_COUNTERS has the LAT$V_STRING bit set to 1, and the string length byte actually contains the length of the entire subblock. The subblock itself is an item list and consists of the item codes listed in Table 4–18.

Table 4–18 Item Codes for Subblock LAT$_ITM_COUNTERS

Item Code	Meaning
LAT$_ITM_CTPRT_LCL	Indicates number of local accesses to port.
LAT$_ITM_CTPRT_SLCA	Indicates number of solicitations accepted.
LAT$_ITM_CTPRT_SLCR	Indicates number of solicitations rejected.
LAT$_ITM_CTPRT_ISOLA	Indicates number of incoming solicitations accepted.
LAT$_ITM_CTPRT_ISOLR	Indicates number of incoming solicitations rejected.
LAT$_ITM_CTPRT_FRAMERR	Indicates number of framing errors for named port. Returned in port counter subblock.
LAT$_ITM_CTPRT_PARERR	Indicates number of parity errors for named port. Returned in port counter subblock.
LAT$_ITM_CTPRT_OVERRUN	Indicates number of data overruns for named port. Returned in port counter subblock.
LAT$_ITM_PASSWORD_FAILURES	Indicates password failures.

4.22.3.2 LAT SENSEMODE Item Code for Node Counters

Table 4–19 lists the LAT SENSEMODE item code for node counters.

Table 4–19 LAT SENSEMODE Item Code for Node Counters

Item Code	Meaning
LAT$_ITM_CTNOD_XSTR (REMOTE)	Indicates that the node attempted to start up too many LAT sessions for a specific virtual circuit.

4.22.3.3 LAT SENSEMODE Item Codes for Protocol Errors

Table 4–20 lists the LAT SENSEMODE item codes for protocol errors.

Table 4–20 LAT SENSEMODE Item Codes for Protocol Errors

Item Code	Meaning
LAT$_ITM_CTPRO_INVCLASS (LOCAL)	Indicates the number of times a LAT message was received with an invalid service class specified in that message (local node only).
LAT$_ITM_CTPRO_EXCSTART (LOCAL)	Indicates that a remote node attempted to start up too many LAT sessions. When a virtual circuit is started between 2 LAT nodes, the maximum number of sessions on that virtual circuit is negotiated. If the master node attempts to create more sessions than the maximum number of sessions on a virtual circuit, then the operating system rejects the excess connections and increments this counter.

4.22.3.4 LAT SENSEMODE Item Code for the Service Counters Subblock

Table 4–21 lists the LAT SENSEMODE item code for the service counters subblock.

Table 4–21 LAT SENSEMODE Item Code for the Service Counters Subblock

Item Code	Meaning
LAT$_ITM_PASSWORD_ FAILURE	Indicates password failures.

4.22.3.5 LAT SENSEMODE Item Codes for Port Entities

Table 4–22 lists LAT SENSEMODE item codes for port entities.

Table 4–22 LAT SENSEMODE Item Codes for Port Entities

Item Code	Meaning
LAT$_ITM_NODE_QUEUE_ POSITION	Indicates current node queue position for connect request. Returned during SENSEMODE of port entity.
LAT$_ITM_SERVICE_QUEUE_ POSITION	Indicates current service queue position for connect request. Returned during SENSEMODE of port entity.

4.22.3.6 LAT SENSEMODE Item Codes for Queue Entries

Table 4–23 lists the LAT SENSEMODE item codes for queue entries.

Table 4–23 LAT SENSMODE Item Codes for Queue Entries

Item Code	Meaning
LAT$_ITM_QUEUED_ENTRY_ ID (SUMMARY)	Indicates by string the queue entry ID name.
LAT$_ITM_NODE_QUEUE_ POSITION (SUMMARY)	Indicates current position of entry in node-wide queue.
LAT$_ITM_SERVICE_QUEUE_ POSITION (SUMMARY)	Indicates current position of entry in service-wide queue.
LAT$_ITM_NODE_NAME (SUMMARY)	Indicates where remote node name queue entry came from.
LAT$_ITM_SERVICE_NAME (SUMMARY)	Indicates the target service name the queue entry is queued to (if specified).
LAT$_ITM_PORT_NAME (SUMMARY)	Indicates the target port name the entry is queued to (if specified).
LAT$_ITM_LINK_NAME	Returns the link name the queued request is active on.
LAT$_ITM_DATALINK_ ADDRESS	Returns remote node that issued request's datalink address.

For more detailed information, see the *OpenVMS I/O User's Reference Manual.*

4.22.4 LAT SETMODE Argument

The LAT SETMODE $QIO function accepts a new argument for the P3 argument. The new argument is **Queue Entry** (LAT$C_ENT_QUEUE_ENTRY). The **Queue Entry** argument indicates queue entry entities. When these entities are used, the only valid SETMODE operation is delete.

For more detailed information, see the *OpenVMS I/O User's Reference Manual.*

4.22.5 LAT SETMODE Item Codes

This section describes the item codes for node entities, service type, and port type.

4.22.5.1 LAT SETMODE Item Codes for Node Entities

The following LAT SETMODE item codes for node entities are only returned if the specified node entity is the local node. Table 4–24 lists the item codes that may be set for the LAT$C_ENT_NODE entity type.

Table 4–24 LAT SETMODE Item Codes for Node Entities

Item Code	Meaning
LAT$_ITM_HI_CIRCUITS	Indicates the highest number the resource attained since the host was initialized for LAT connections to node.
LAT$_ITM_CUR_CIRCUITS	Indicates current count of active connections to node.
LAT$_ITM_MAX_CIRCUITS	Indicates maximum allowed virtual circuits to node.
LAT$_ITM_HI_SESSIONS	Indicates highest number the resource attained since the host was initialized for LAT sessions.
LAT$_ITM_CUR_SESSIONS	Indicates current number of active sessions.
LAT$_ITM_MAX_SESSIONS	Indicates maximum possible sessions.

(continued on next page)

Table 4–24 (Cont.) LAT SETMODE Item Codes for Node Entities

Item Code	Meaning
LAT$_ITM_HI_OUT_QUEUE	Indicates highest number the resource attained since the host was initialized of outgoing queued connect requests.
LAT$_ITM_CUR_OUT_QUEUE	Indicates current count of outgoing queued connect requests.
LAT$_ITM_MAX_OUT_QUEUE	Indicates maximum number of simultaneous outgoing queued connect requests.
LAT$_TIM_HI_IN_QUEUE	Indicates highest number the resource attained since the host was initialized of incoming queued requests.
LAT$_ITM_CUR_IN_QUEUE	Indicates current number of entries in the incoming connect queue.
LAT$_ITM_MAX_IN_QUEUE	Indicates maximum number of entries allowed on the incoming connect queue.
LAT$_ITM_HI_SAMS_QUEUED	Indicates highest number the resource attained since the host was initialized of outstanding, unprocessed service announcement messages by LAT ancillary control process (LATACP).
LAT$_ITM_CUR_SAMS_ QUEUED	Indicates current number of outstanding, unprocessed service announcement messages on LATACP's queue.
LAT$_ITM_MAX_SAMS_ QUEUED	Indicates maximum number of outstanding, unprocessed service announcement messages allowed on LATACP's queue. If this limit is ever reached, subsequent service announcement messages will not be delivered or processed by LATACP.
LAT$_ITM_HI_SOL_QUEUED	Indicates highest number the resource attained since the host was initialized of outstanding, unprocessed solicit information messages by LATACP.
LAT$_ITM_CUR_SOL_QUEUED	Indicates current number of outstanding, unprocessed solicit information messages on LATACP's queue.
LAT$_ITM_MAX_SOL_ QUEUED	Indicates maximum number of outstanding, unprocessed solicit information messages allowed on LATACP's queue. If this limit is ever reached, subsequent solicit information messages will not be delivered or processed by LATACP.
LAT$_ITM_HI_AVAIL_SVCS	Indicates highest number the resource attained since the host was initialized of number of available services in LATACP database.
LAT$_ITM_CUR_AVAIL_SVCS	Indicates count of currently available LAT services in LATACP database.
LAT$_ITM_MAX_AVAIL_SVCS	Indicates maximum number of available services possible in LATACP database.
LAT$_ITM_HI_REACH_NODES	Indicates highest number the resource attained since the host was initialized of reachable nodes in LATACP database.
LAT$_ITM_CUR_REACH_ NODES	Indicates current number of reachable nodes in LATACP in database.
LAT$_ITM_MAX_REACH_ NODES	Indicates maximum number of nodes allowed in LATACP database.
LAT$_ITM_HI_LCL_SVCS	Indicates highest number the resource attained since the host was initialized of locally offered services.
LAT$_ITM_CUR_LCL_SVCS	Indicates current count of locally offered services.
LAT$_ITM_MAX_LCL_SVCS	Indicates maximum number of locally offered services.
LAT$_ITM_DISCARDED_ NODES	Indicates number of discarded service announcement messages.

(continued on next page)

Table 4–24 (Cont.) LAT SETMODE Item Codes for Node Entities

Item Code	Meaning
LAT$_ITM_CTPRO_INVCLASS	Indicates the number of times a LAT message was received with an invalid service class specified in that message (local node only). Returned in counter subblock.
LAT$_ITM_CTPRO_EXCSTART	Indicates that a remote node attempted to start up too many LAT sessions. When a virtual circuit is started between 2 LAT nodes, the maximum number of sessions on that virtual circuit is negotiated. If the master node attempts to create more sessions than the maximum number of sessions on a virtual circuit, then the operating system rejects the excess connections and increments this counter. Returned in the protocol error counter subblock.
LAT$_ITM_CTNOD_XSTR	Indicates that the node attempted to start up too many LAT sessions for a specific virtual circuit. This counter only appears in the node counter block for remote nodes. Returned in the node counter subblock.
LAT$_ITM_SERVICE_CLASSES	Indicates returned service class bit mask for supported service classes on node. It is returned for both local and remote nodes. If service class 1 is enabled, then bit <1> is set in this mask. When bit setting equals 1, this indicates the corresponding service class for that bit is enabled. That is, when bit <3> equals 1, then service class 3 is enabled.

4.22.5.2 LAT SETMODE Item Code for Service Type

Table 4–25 lists the LAT SETMODE item codes for the LAT$C_ENT_SERVICE service type.

Table 4–25 LAT$C_ENT_SERVICE Item Codes

Item Code	Meaning
LAT$C_ST_LIMITED	Indicates that the service is limited.
LAT$_ITM_PASSWORD	Indicates that if a value of LAT$C_ENABLED is set, then the service is password protected. Indicates that if a value of LAT$C_DISABLED is set, then the service is not password protected.
LAT$_ITM_LIM_PORT_BLOCK	Indicates a subblock contained in an item list that has a list of limited ports that are associated with the named service. This subblock may be repeated several times; that is, once for each limited LAT device associated with the specified service.

4.22.5.3 LAT SETMODE Item Code for Port Type

Table 4–26 lists the LAT SETMODE item codes for LAT$C_ENT_PORT port type.

Table 4–26 LAT$C_ENT_PORT Item Codes

Item Code	Meaning
LAT$C_PT_LIMITED	Indicates that the port type is limited.
LAT$_ITM_DIALUP	Used to set whether a LAT device indicates to a remote node that the connection is coming from a dialin source. Possible values = LAT$C_ ENABLED or LAT$C_DISABLED.
LAT$_ITM_AUTOPROMPT	Used to indicate whether a connect request has autoprompt enabled. Possible values = LAT$C_ENABLED or LAT$C_DISABLED.

For more detailed information, see the *OpenVMS I/O User's Reference Manual.*

4.23 $SETIME and DECdts

The $SETIME system service and distributed time synchronization service (DECdts) are no longer system specific to OpenVMS but are supported on both OpenVMS VAX and OpenVMS AXP systems.

For a description of the feature, see the *OpenVMS Programming Concepts Manual*.

4.24 SS$_PAGRDERR Condition

The signal argument of the SS$_PAGRDERR exception condition provides the actual I/O failure status that caused the page read error, as shown in Table 4–27.

Table 4–27 SS$_PAGRDERR Condition

Condition Name	Explanation	
SS$_PAGRDERR	Type:	Fault
	Description:	Read error occurred during an attempt to read a faulted page from disk.
	Arguments:	1. I/O failure status.
		2. Virtual address of referenced page.

For more detailed information, see the *OpenVMS Programming Concepts Manual*.

4.25 LAN Programming Features

This section contains information about new local area network (LAN) programming features.

4.25.1 LAN Devices

Table 4–28 shows the devices, buses, drivers, device names, device types, and media that are supported on OpenVMS AXP Version 6.1. Note that the c in the device name indicates the controller letter, and that LAN devices always have a device class of DC$_SCOM.

Table 4–28 New LAN Devices, Drivers, and Busses

Device	Bus	Driver	Device Name	Device Type	Medium
‡Digital Equipment TRNcontroller 700 (DETRA)	TURBOchannel	ICDRIVER	ICc0:	DT$_IC_DETRA	Token Ring
‡Proteon PROnet-4/16 EISA NIC (DW300)	EISA	IRDRIVER	Rc0:	DT$_IR_DW300	Token Ring
‡Digital Ethernet EISA Bus Adapter (DE422)	EISA	ERDRIVER	ERc0:	DT$_ER_DE422	Ethernet
‡Digital Ethernet EISA Bus Adapter (DE425)	EISA	ERDRIVER	ERc0:	DT$_ER_DE425	Ethernet
‡Digital Ethernet PCI Bus Adapter (TULIP)	PCI	EWDRIVER	EWc0:	DT$_EW_ TULIP	Ethernet
†Digital Equipment FDDIcontroller (DEFQA)	Q–bus	FQDRIVER	FQc0:	DT$_FQ_ DEFQA	FDDI
‡Digital Equipment FDDIcontroller (DEFEA)	EISA	FRDRIVER	FRc0:	DT$_FR_ DEFEA	FDDI
‡Digital Equipment FDDIcontroller (DEFAA)	FUTUREBUS	FADDRIVER	FAc0:	DT$_FA_ DEFAA	FDDI

†VAX specific.
‡AXP specific.

For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.2 Token Ring Characteristics

The LAN controllers are direct-memory-access (DMA) devices that, along with additional external hardware, implement the Ethernet, FDDI, or Token Ring specifications.

The Token Ring controllers use either shielded or unshielded twisted pairs of wire to access the ring. Note that it is difficult to connect a Token Ring LAN directly bridged to any other LAN. However, routing protocols to other LANs work easily.

For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.3 LAN Token Ring Functional Address Mapping

With OpenVMS AXP Version 6.1, except for the global broadcast address (FF-FF-FF-FF-FF-FF), Token Ring hardware does not support the 802 standard group LAN address mechanism. Instead, it uses **functional addresses**. These functional addresses are locally administered group addresses (multicast addresses). The first two bytes of the address are always 03-00 (canonical format), and the remaining four bytes contain a bit mask that specifies which of the 32 possible combination masks is being described.

Because most OpenVMS LAN applications use standard multicast addresses, a mechanism has been designed to map functional addresses to globally and locally administered multicast addresses. This allows applications to use the same multicast addresses that are used in the other LAN mediums.

Table 4–29 shows the address mappings for the OpenVMS Token Ring drivers.

Table 4–29 Address Mappings of Token Ring Drivers

Multicast Address	Functional Address	Bit-Reversed	Description
09-00-2B-00-00-04	03-00-00-00-02-00	C0:00:00:00:40:00	ISO ALL ES
09-00-2B-00-00-05	03-00-00-00-01-00	C0:00:00:00:80:00	ISO ALL IS
CF-00-00-00-00-00	03-00-00-08-00-00	C0:00:00:10:00:00	Loopback Assistance
AB-00-00-01-00-00	03-00-02-00-00-00	C0:00:40:00:00:00	Mop Dump/Load
AB-00-00-02-00-00	03-00-04-00-00-00	C0:00:20:00:00:00	Mop Remote Console
AB-00-00-03-00-00	03-00-08-00-00-00	C0:00:10:00:00:00	DNA L1 Routers
09-00-2B-02-00-00	03-00-08-00-00-00	C0:00:10:00:00:00	DNA L2 Routers
09-00-2B-02-01-0A	03-00-08-00-00-00	C0:00:10:00:00:00	Phase IV prime Routers
AB-00-00-04-00-00	03-00-10-00-00-00	C0:00:08:00:00:00	DNA Endnodes
09-00-2B-02-01-0B	03-00-10-00-00-00	C0:00:08:00:00:00	Phase IV Prime Unknown
09-00-2B-00-00-07	03-00-20-00-00-00	C0:00:04:00:00:00	PCSA NETBIOS Emulatn
09-00-2B-00-00-0F	03-00-40-00-00-00	C0:00:02:00:00:00	LAT Service Adver
09-00-2B-02-01-04	03-00-80-00-00-00	C0:00:01:00:00:00	LAT Service Solicit
09-00-2B-02-01-07	03-00-00-02-00-00	C0:00:00:40:00:00	LAT Xwin Server Solicit
09-00-2B-04-00-00	03-00-00-04-00-00	C0:00:00:20:00:00	LAST
09-00-2B-02-01-00	03-00-00-00-08-00	C0:00:00:00:10:00	DNA Name Server Adver
09-00-2B-02-01-01	03-00-00-00-10-00	C0:00:00:00:08:00	DNA Name Server Solicit
09-00-2B-02-01-02	03-00-00-00-20-00	C0:00:00:00:04:00	DNA Time Service
03-00-00-00-00-01	03-00-00-00-00-01	C0:00:00:00:00:80	NETBUI Emulation

If an application needs to change or add mappings, QIOs exist for performing such operations. If the system or network manager has a requirement regarding mapping of the functional addresses, the LAN Control Program (LANCP) utility may be used to manage the mapping by running **sys$system:lancp**. The following example maps the multicast address AB-01-01-01-02-03 to functional address 03-00-00-01-00-00 on Token Ring device ICA0:.

```
LANCP>SET DEVICE/MAP= -
_LANCP>(MULTICAST=AB-01-01-01-02-03,-
_LANCP>FUNCTIONAL=00-01-00-00) ICA0:
```

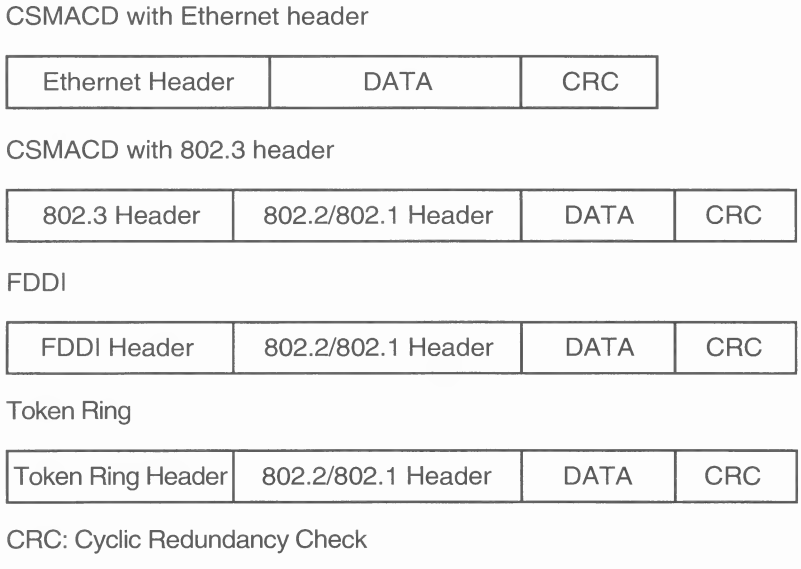
Note that it is possible for more than one multicast address to map to the same functional address. In all cases, the use of the functional address is associated with an individual application's protocol.

For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.4 LAN Frame Formats

Figure 4–3 shows the Token Ring frame format.

Figure 4–3 LAN Frame Formats

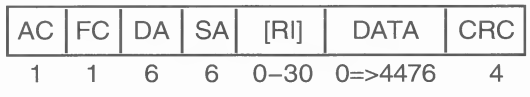


For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.4.1 LAN Token Ring Frames

Figure 4–4 shows the format of Token Ring frames.

Figure 4–4 Token Ring Frame Format



AC: Access Control contains priority for the frame.
FC: Frame Control contains the type of frame.
DA: Destination Address
SA: Source Address
RI: Optional Routing Information. Only valid with packets that are source routed.
DATA: User's data
CRC: Cyclic Redundancy Check

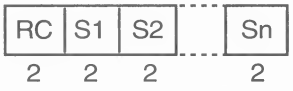
ZK–6679A–GE

For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.4.2 LAN Token Ring Source Routing Header

Figure 4–5 shows the field of the source routing header.

Figure 4–5 Source Routing Field



ZK–6680A–GE

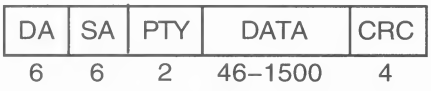
For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.5 LAN Format Parameter for Token Ring

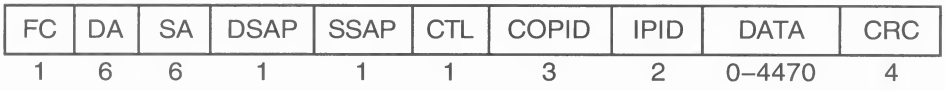
Figure 4–6 shows frames used with the Ethernet format.

Figure 4–6 Frames with Ethernet Format

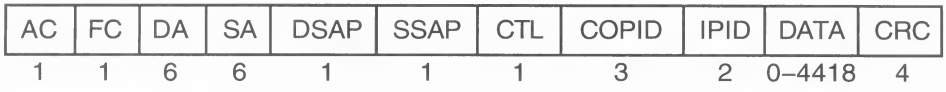
For CSMA/CD



For FDDI



For Token Ring



DSAP: SNAP SAP (hex AA)
SSAP: SNAP SAP
CTL: UI (hex 03)
COPID: Mapped Ethernet (00–00–00)
IPID, PTY: Protocol Type

ZK–3905A–GE

Figure 4–7 shows the frames used with the 802 format.

Figure 4–7 Frames with 802 Format

For CSMA/CD

DA	SA	LEN	DSAP	SSAP	CTL	DATA	CRC
6	6	2	1	1	1–2	N	4

For FDDI

FC	DA	SA	DSAP	SSAP	CTL	DATA	CRC
1	6	6	1	1	1–2	Z	4

For Token Ring

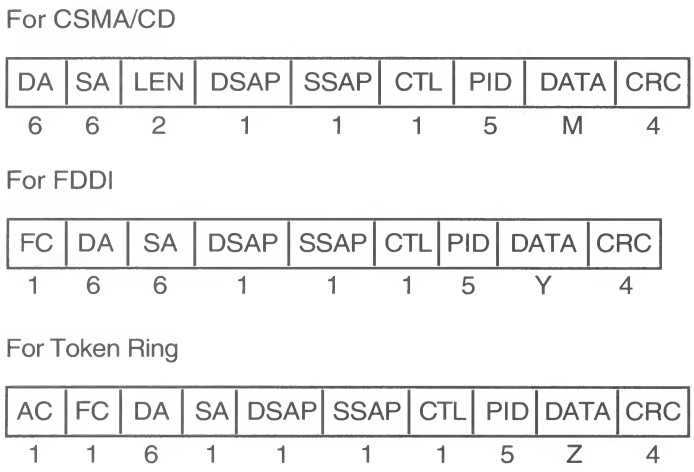
AC	FC	DA	SA	DSAP	SSAP	CTL	DATA	CRC
1	1	6	6	1	1	1–2	Y	4

DSAP: Destination Service Access Point
SSAP: Source Service Access Point
CTL: Control Field
DATA length 43 <= N <= 1497 (1 byte CTL)
42 <= N <= 1496 (2 byte CTL)
DATA length 0 <= Z <= 4475 (1 byte CTL)
0 <= Z <= 4474 (2 byte CTL)
DATA length 0 <= Y <= 4423 (1 byte CTL)
DATA length 0 <= Y <= 4424 (2 byte CTL)

ZK–3902A–GE

Figure 4–8 shows the frames used with the 802E format.

Figure 4–8 Frames with 802E Format


For CSMA/CD
DA SA LEN DSAP SSAP CTL PID DATA CRC
6 6 2 1 1 1 5 M 4
For FDDI
FC DA SA DSAP SSAP CTL PID DATA CRC
1 6 6 1 1 1 5 Y 4
For Token Ring
AC FC DA SA DSAP SSAP CTL PID DATA CRC
1 1 6 1 1 1 1 5 Z 4
DSAP: SNAP SAP (hex AA)
SSAP: SNAP SAP
CTL: UI (hex 03)
PID: User's Protocol Identifier
DATA length 38 <= M <= 1492
DATA length 0 <= Y <= 4470
DATA length 0 <= Z <= 4418
ZK-3903A-GE

For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.6 LAN Protocol Type PID Sharing

With OpenVMS AXP Version 6.1, the 802E format allows user's protocol identifier (PID) sharing. PIDs are usually nonshareable. The problems inherent in sharing a PID include the multiplexing and demultiplexing of messages to and from remote nodes. However, the protocol access parameter (NMA$C_PCLI_ACC) allows a PID to be opened in either of two shareable modes: shared-default (NMA$C_ACC_SHR) and shared-with-destination (NMA$C_ACC_LIM). The LAN drivers also provide the nonshareable exclusive mode (NMA$C_ACC_EXC). The rules and requirements for using each mode are as follows:

- The exclusive mode is the default if no access mode is supplied as a P2 buffer parameter. This mode of operation does not allow the protocol to be shared by other users. Any attempt to start up another protocol of the same type results in an error status of SS$_BADPARAM.
- The shared-with-destination mode is a PID/destination address pairing that allows multiple users to share a PID and to communicate with a different node.

  For a given shared protocol type, there can be many "shared-with-destination" users; each user communicates with a different destination address. Any attempt to start a port with a destination address that is in use results in an error status of SS$_BADPARAM.

When a "shared-with-destination" user passes the set mode P2 buffer, the buffer must contain a destination address in the NMA$C_PCLI_DES parameter. This destination address is used as the destination address in all messages transmitted, and the user receives messages only from this address.

- The shared-default mode is the default user of a shared PID. There can be only one such user for each shared PID. A "shared-default" user does not have to exist if a PID is shared, but there can be no more than one such user per shared PID.

  The "shared-default" user receives all messages for the shared PID, but not for any of the "shared-with-destination" users. The "shared-default" user also receives all messages matching both the shared PID and any multicast address enabled by the "shared-default" user.

  The "shared-default" user can only transmit to multicast addresses and physical addresses that are not enabled by any of the "shared-with-destination" users sharing the same PID.

  If there is no "shared-default" user of a PID, incoming messages from nodes not among the "shared-with-destination" users for that PID are ignored.

For more information, see the *OpenVMS I/O User's Reference Manual.*

4.25.7 LAN Device Information

You can obtain information on controller characteristics by using the Get Device /Volume Information ($GETDVI) system service. (See the *OpenVMS System Services Reference Manual.*) The $GETDVI system service returns controller characteristics when you specify the item code DVI$_DEVCHAR. Table 4–30 lists these characteristics, which are defined by the $DEVDEF macro and in the file SYS$LIBRARY:DEVDEF.H.

Table 4–30 Ethernet Controller Device Characteristics

Ethernet Controller	Device	
	Type	Class Name
TGEC	DT$_EZ_TGEC	DC$_SCOM
DETRA	DT$_IC_DETRA	DC$_SCOM
DW300	DT$_IR_DW300	DC$_SCOM
DE422	DT$_ER_DE422	DC$_SCOM
DE425	DT$_ER_TULIP	DC$_SCOM
TULIP	DT$_EW_TULIP	DC$_SCOM

For more information, see the *OpenVMS I/O User's Reference Manual.*

4.25.8 LAN Read with Token Ring

Read functions directly transfer data from a packet received from another port on the Ethernet into the virtual memory address space of the user process. The operating system provides the following function codes:

- IO$_READLBLK—Read logical block
- IO$_READVBLK—Read virtual block
- IO$_READPBLK—Read physical block

Received messages are buffered in system memory and then copied to the user's buffer when a read operation is performed.

The read functions take the following device- or function-dependent arguments:

- P1—The starting virtual address of the buffer that is to receive data.
- P2—The size of the receive buffer in bytes.
- P5—The address of a buffer where the LAN driver returns packet header information. This is an optional argument. The information returned depends on the packet format enabled with the set mode QIO. The size of the buffer must be 14 bytes for an Ethernet format packet, 16 bytes for an IEEE 802 format packet, and 20 bytes for an 802 extended format packet. Note that the information returned is not the entire packet header but the header information less any length or size fields. The IOSB, if specified, is where the packet length information is returned. For FDDI, if received access control (RAC) is on, then 1 byte must be added to these sizes.

With OpenVMS AXP Version 6.1, Token Ring requires that this buffer be at least 54 bytes long due to a possible variable length source routing header.

Table 4–31 shows the maximum user data sizes that can be received.

Table 4–31 Maximum User Data Sizes

Packet Format	CSMA/CD	FDDI	Token Ring
Ethernet format without padding	1500	4470	4418
Ethernet format with padding	1498	4468	4416
802 format with 1-byte CTL field	1497	4475	4423
802 format with 2-byte CTL field	1496	4474	4422
802E format	1492	4470	4418

When using Token Ring with received access control (RAC) on, the first byte of the P5 buffer contains the frame control (FC) field.

For more information, see the *OpenVMS I/O User's Reference Manual.*

4.25.9 LAN Write with Token Ring

Write functions provide for the direct transfer of data from the virtual memory address space of the user process to the communications medium. The operating system provides the following function codes:

- IO$_WRITELBLK—Write logical block
- IO$_WRITEVBLK—Write virtual block
- IO$_WRITEPBLK—Write physical block

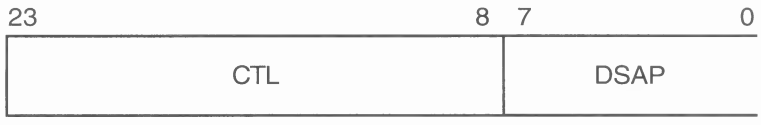
Transmitted messages are copied from the buffer of the requesting process to a system buffer for transmission.

The write function takes the following device- or function-dependent arguments:

- P1—The starting virtual address of the buffer containing the data to be transmitted.
- P2—The size of the buffer in bytes.

- P4—The address of a quadword that points to a buffer that contains the DSAP and CTL field values (optional). The first longword is the buffer length; the second longword is the address of the buffer. This argument is used only for ports with the 802 packet format. Figure 4–9 shows the format of the buffer.

Figure 4–9 Format of the DSAP and CTL Buffer

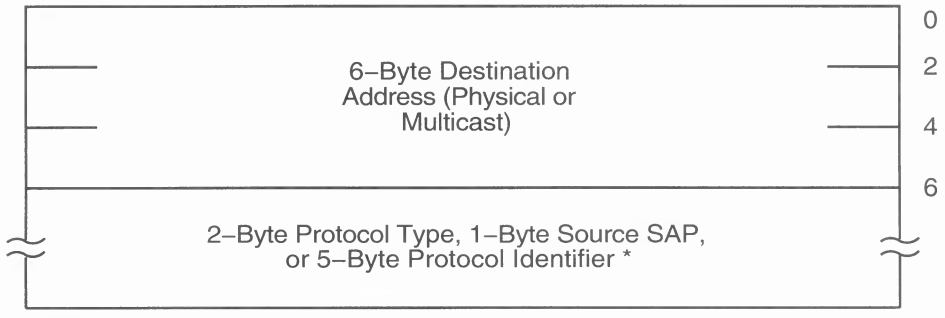


ZK–4801–GE

- P5—The address of a 6-byte buffer that contains the destination address (either physical or multicast). For FDDI, if transmit frame control (XFC) is specified as zero on startup, the first byte of the P5 buffer contains the low-order 3 bits of the FC field to be transmitted.

  If the device is in promiscuous mode (NMA$C_PCLI_PRM; see Table 4–33), you must pass a larger buffer with additional information positioned after the destination address. For Ethernet packet format, the buffer must be 8 bytes with the 2-byte protocol type following the destination address. For 802 packet format, the buffer must be 7 bytes with the 1-byte source service access point (SAP) following the destination address. For 802 extended packet format, the buffer must be 11 bytes with the 5-byte protocol identifier following the destination address. The individual source SAP cannot be a group SAP or the subnet access point (SNAP) SAP. Figure 4–10 shows the format of the P5 buffer. For FDDI with XFC specified as zero on startup, 1 byte must be added to these sizes for the FC field.

Figure 4–10 Write Function P5 Buffer



*Only if the channel is in promiscuous mode.

ZK–1211–GE

Table 4–32 lists the maximum user message sizes that can be specified by P2.

Table 4–32 Maximum Message Sizes

Packet Format	CSMA/CD	FDDI	Token Ring
Ethernet format without padding	1500	4470	4418
Ethernet format with padding	1498	4468	4416
802 format with 1-byte CTL field	1497	4475	4423
802 format with 2-byte CTL field	1496	4474	4422
802E format	1492	4470	4418

For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.10 LAN Set Mode and Set Characteristics for Token Ring

The operating system provides the following two function codes:

IO$_SETMODE
IO$_SETCHAR

Other than the privilege check, these two function codes are treated the same by the LAN drivers. This section refers to the IO$_SETMODE function code only, even though applications can use either function code.

The set mode function code is used to perform many different functions. These different functions are distinguished by the modifiers set with the function code. The LAN drivers support the following set mode requests for Token Ring:

- IO$M_SETMODE!IO$M_UPDATE_MAP—Update functional address mapping table
- IO$M_SETMODE!IO$M_ROUTE— Update source routing cache table

For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.10.1 LAN Set Controller Mode

Once a port is created using the $ASSIGN system service, you can set the port attributes and start the port. Note that in most cases only IO$_SETMODE!IO$M_CTRL!IO$M_STARTUP is issued because it sets the port attributes and starts the port with one request. The IO$_SETMODE!IO$M_CTRL function code is most often used to modify port attributes after the port has been started.

If the function modifier IO$M_STARTUP is specified, the LAN port is started. If IO$M_STARTUP is not specified, the specified characteristics are modified.

This function takes the following device- or function-dependent argument:

- P2—The address of a quadword descriptor for an extended characteristics buffer. The first longword of the descriptor is the buffer length; the second longword is the address of the buffer. The P2 argument is optional.

The P2 buffer consists of a series of 6-byte or counted string entries. The first word of each entry contains the parameter identifier (ID) of an attribute, followed by either a longword that contains one of the (binary) values that can be associated with the parameter ID or a counted string. Counted strings consist of a word that contains the size of the character string followed by the character string. Table 4–33 is an alphabetic listing of the parameter IDs and values that can be specified in the P2 buffer for Token Ring. These parameter IDs are

applicable to all LAN controllers, except where otherwise noted. The $NMADEF macro defines these values. The $NMADEF macro is included in the macro library SYS$LIBRARY:LIB.MLB. Table 4–34 lists the parameters that can be used with each of the packet formats and indicates which are required and which are optional.

Table 4–33 P2 Attributes

Parameter ID	Meaning
NMA$C_PCLI_ACC	Valid for ports using Ethernet and 802E.
NMA$C_PCLI_CRC	For the DEQNA, the DELQA, and the Token Ring devices, the NMA$C_PCLI_CRC parameter cannot be turned off.
NMA$C_PCLI_DES	Should not be specified on a port where the 802 packet format is selected (NMA$C_PCLI_FMT is set to NMA$C_LINFM_802). For 802 packet format the concept of shared protocol type is handled by using group SAPs.
NMA$C_PCLI_ILP	Internal loopback mode. This optional parameter places the device in internal loopback mode (not for the DEUNA, DEQNA, or DELQA devices). This parameter has no effect for Token Ring devices.
NMA$C_PCLI_PRM	Promiscuous mode (optional). Some Token Ring devices do not support real promiscuous access to the Token Ring.
NMA$C_PCLI_RAC	Receive access control (Token Ring only). This optional parameter specifies whether the application receives a copy of the access control (AC) field for each Token Ring frame received. It is passed as a longword value. It must be passed with one of the following values: • NMA$C_STATE_ON — Application gets a copy of the AC for each Token Ring frame received. • NMA$C_STATE_OFF — Application does not get a copy of AC for each Token Ring frame received. The AC is returned in the P5 buffer. The P5 buffer size for Token Ring should always be a minimum of 54 bytes. This is due to the variable size of the Token Ring header.
NMA$C_PCLI_SRMODE	Sets the source routing (SR) mode for the $QIO user (Token Ring only). This optional parameter allows the application to perform the source routing discovery. It must be passed with one of the following values: • NMA$C_SR_TRANSPARENT — Application source routing is transparent. This is the default when this parameter is not specified. • NMA$C_SR_SELF — This shuts off the automatic route discovery exploration messages for this user. The $QIOs exist to further manipulate the source routing cache. Digital recommends that applications use the NMA$C_SR_TRANSPARENT mode.
NMC$C_PCLI_XAC	Transmit access control (Token Ring only). This is an optional parameter that enables applications to control the setting of the priority bits in the access control (AC) for frames being transmitted in a $QIO write operation. When set to the wanted value, all subsequent transmits use this AC value.

For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.10.2 LAN Set Mode Parameters for Packet Formats for Token Ring

Table 4–34 summarizes the use of the set mode parameters for the Ethernet, 802, and 802 extended (802E) packet formats.

Table 4–34 Set Mode Parameters for Packet Formats

Parameter ID	Ethernet	IEEE 802	802E
RAC	OPT	OPT	OPT
SRMODE	OPT	OPT	OPT
XAC	OPT	OPT	OPT

Legend

OPT—Optional. This parameter is optional for this packet format; it may be specified.

For more information, see the *OpenVMS I/O User's Reference Manual.*

4.25.10.3 IO$M_SET_MAC Functional Modifier to IO$M_SETMODE

The IO$M_SET_MAC qualifier, when used with IO$_SETMODE, is used to set medium specific parameters. To set the Token Ring parameters, the PHY_IO privilege is required. Table 4–35 shows the parameters that may be set.

Table 4–35 Medium Specific Parameters of IO$M_SET_MAC

Parameter ID	Meaning
NMA$C_PCLI_RNG_SPD	Sets the speed of the ring (Token Ring only). This longword may be either: NMA$C_LINRNG_FOUR—Used for 4 Mbps rings. NMA$C_LINRNG_SIXTN—Used for 16 Mbps rings. The default is NMA$C_LINRNG_SIXTN.
NMA$C_PCLI_LINEMEDIA	Sets the connection media type for the Token Ring adapter (Token Ring only). The following are valid values for this longword parameter: NMA$C_MEDIA_STP NMA$C_MEDIA_UTP The default is NMA$C_MEDIA_STP.
NMA$C_PCLI_ETR	Controls the Early Token Release feature of the Token Ring hardware (Token Ring only). This feature can greatly improve throughput, and is only valid on 16 Mbps rings. The values for this longword parameter are NMA$C_STATE_ON or NMA$C_STATE_OFF. The default is NMA$C_STATE_ON.
NMA$C_PCLI_MONCONTEND	Specifies whether the controller participates in the monitor contention process when another adapter detects the need for contention and initiates the process (Token Ring only). The values for this longword parameter are NMA$C_STATE_ON or NMA$C_STATE_OFF. The default is NMA$C_STATE_OFF.
NMA$C_PCLI_CACHE_ENT	The number of source routing (SR) entries to make available for caching (Token Ring only). The default is 200, minimum is 20, and maximum is 2000. Each cache entry consumes 64 bytes.

(continued on next page)

Table 4–35 (Cont.) Medium Specific Parameters of IO$M_SET_MAC

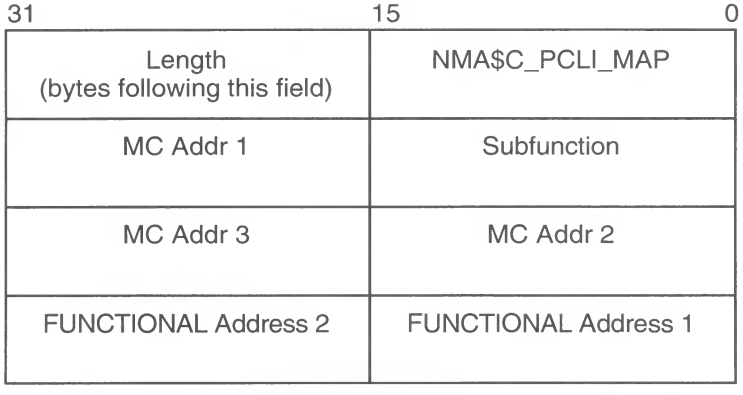
Parameter ID	Meaning
NMA$C_PCLI_ROUTEDIS	The source routing discovery timer (Token Ring only). This is the amount of seconds to wait after the transmission of ring explorer packets before declaring the route of a path to be unknown. The default is 2 seconds, minimum is 1, and maximum is 255.
NMA$C_PCLI_A_TIM	The source routing aging timer (Token Ring only). After traffic is neither received from nor sent to a given node for this number of seconds, the entry is marked stale. After the entry is marked stale, rediscovery is required to communicate with the node. The default is 60 seconds, minimum is 1, and maximum is 65535.
NMA$C_PCLI_SRC_ROU	Enables and Disables Source Routing (Token Ring only). The values for this longword parameter are NMA$C_LINSRC_ENA or NMA$C_ LINSRC_DIS. The default is NMA$C_LINSRC_DIS.
NMA$C_PCLI_AUTH_PR	Specifies the highest priority that a user may transmit a frame (Token Ring only). The priority is set within the NMA$C_PCLI_XAC parameter. The default for this parameter is 3, minimum is 0, and maximum is 6.

For more information, see the *OpenVMS I/O User's Reference Manual.*

4.25.10.4 IO$M_UPDATE_MAP Functional Modifier to IO$_SETMODE with Token Ring

With OpenVMS AXP Version 6.1 using Token Ring only, the IO$M_UPDATE_ MAP qualifier, when used with IO$_SETMODE, manipulates the adapter's functional address mapping table. Figure 4–11 shows the format of the P2 buffer for this operation. This QIO requires PHY_IO privilege.

Figure 4–11 Format of IO$M_UPDATE_MAP Setmode P2 Buffer (AXP Only)



ZK–6791A–GE

The subfunction is one of the following:

- NMA$C_MAP_CHANGE—This function adds or changes a mapping in the functional address table. If the specified multicast entry does not exist, an entry is created with the specified functional address mask. If the specified multicast entry does exist, the corresponding functional address mask is changed to the specified mask. All users who currently have the multicast enabled when the functional mask is changed will automatically update the functional address table as part of this operation.

Possible errors returned includes the following:

SS$_DEVICEFULL—This error indicates that there is insufficient space in the mapping table to complete the request. The multicast to functional address mapping table has 200 entries.

- NMA$C_MAP_DELETE—This function deletes the specified MC address in the table. For this function, the functional address mask is not required to pass the P2 buffer. If the functional address mask is passed, its contents are ignored.

  The following possible error is returned:

  SS$_BADPARAM — This error indicates that the specified multicast address cannot be found in the table.

The following example maps multicast address AB-01-01-01-02-03 to the functional address 03-00-00-01-00-00 for device ICA0:.

```
LANCP>SET DEVICE/MAP= -
_LANCP>(MULTICAST=AB-01-01-01-02-03, -
_LANCP>FUNCTIONAL=00-01-00-00) ICA0:
```

The following example deletes the mapping of the multicast address of AB-01-01-01-02-03 for the device ICA0:.

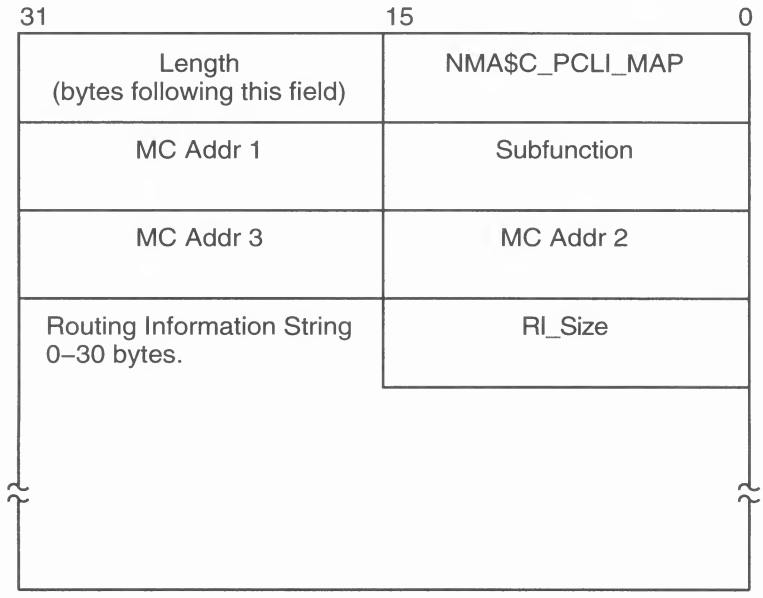
```
LANCP>SET DEVICE/NOMAP=(MULTICAST=AB-01-01-01-02-03) ICA0:
```

For more information, see the *OpenVMS I/O User's Reference Manual.*

4.25.10.5 IO$M_ROUTE Functional Modifier to IO$_SETMODE with Token Ring

With OpenVMS AXP Version 6.1 using Token Ring only, the IO$M_ROUTE qualifier, when used with IO$_SETMODE, manipulates the source routing cache table. This command is successful only when source routing is enabled. Source routing is enabled with the set mac qualified set mode QIO. Figure 4–12 shows the format of the P2 buffer. This QIO requires PHY_IO privilege.

Figure 4–12 Format of the IO$M_ROUTE P2 Buffer (AXP Only)



ZK–6792A–GE

The subfunction is one of the following:

- NMA$C_SR_ADD—This function adds or changes a source routing cache entry. It enters the LAN address into the table with the enclosed routing information. If RI_size is passed as 0, the entry is created (or modified) to be in the EXPLORING state (this is useful for users that are doing their own source routing). If the RC 'Lth' field is zero, the LAN address is entered in the table as being in the LOCAL state.

  Possible errors returned include:

  SS$_INSFMEM—The source routing cache is full.
  SS$_BADPARAM—An invalid RI string was passed or invalid sizes were passed.
  SS$_IVMODE—Source routing is not enabled.

- NMA$C_SR_DELETE—This function deletes a source routing cache entry. The RI_size and the Routing Information String are not required for this QIO. If one or both of the fields are passed for this operation, they are ignored. The result of this command is to put the entry into the DELETED state. When the entry goes into the deleted state, it is deleted within 10 minutes.

  The following is a possible error returned:

  SS$_BADPARAM—The requested entry could not be found.

For more information, see the *OpenVMS I/O User's Reference Manual.*

4.25.11 Sense Mode and Sense Characteristics for Token Ring

The sense mode function returns the port attributes in the specified buffers. These attributes include the device characteristics described in Table 4–33.

The following combinations of function code and modifier for Token Ring are provided:

- IO$_SENSEMODE!IO$M_SHOW_MAP—Returns current functional address to multicast address mapping
- IO$_SENSEMODE!IO$M_SHOW_ROUTE—Returns current source routing cache table

Table 4–36 shows the sense-mode P2 buffer parameter ID's.

Table 4–36 Parameter ID's for P2 Buffer

Parameter ID	Meaning
NMA$C_PCLI_FCA	List of the currently enabled functional addresses (Token Ring only). Each 32-bit entry corresponds respectively with the items returned under NMA$C_PCLI_MCA.
NMA$C_PCLI_MBS	Maximum packet length. NMA$C_PCLI_MBS is a longword, read-only parameter. The value returned reflects the largest data packet that the application can receive for its packet format and type of LAN, measured in bytes. The values include:

Packet format	CSMA/CD	FDDI	Token Ring
Ethernet	1500	4470	4418
Ethernet with PAD	1498	4468	4416
802	1497	4475	4423
802E	1492	4470	4418

For more information, see the *OpenVMS I/O User's Reference Manual.*

4.25.11.1 IO$M_SENSE_MAC Functional Modifier to IO$_SENSEMODE

The IO$M_SENSE_MAC qualifier, when used with IO$_SENSEMODE, returns the parameters specified in Section 4.25.10.3. In addition to the set mac parameters, Table 4–37 shows the returns of the following parameters.

Table 4–37 Parameters of IO$M_SENSE_MAC

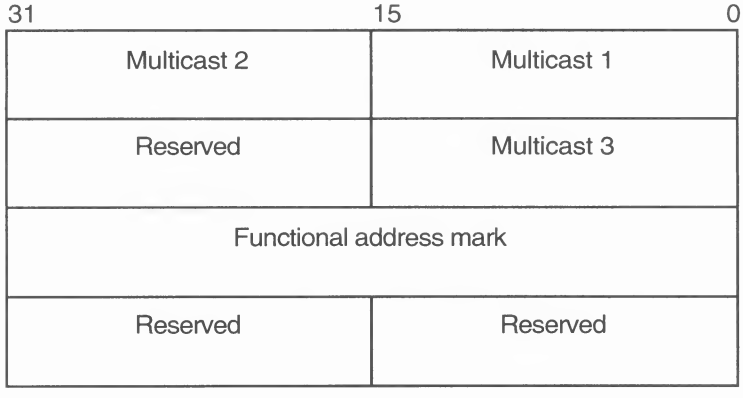
Parameter ID	Meaning
NMA$C_PCLI_UNA	Upstream neighbor's address (FDDI and Token Ring). This is a string parameter specifying the 6 byte LAN address of the upstream neighbor. Not all devices may support this feature.
NMA$C_PCLI_RNG_NUM	The longword value contains the ring number that the controller is running on (Token Ring only). It is valid only for a controller that is started and only valid for rings that have a ring parameter server that is configured for providing this information.

For more information, see the *OpenVMS I/O User's Reference Manual.*

4.25.11.2 IO$M_SHOW_MAP Functional Modifier to IO$_SENSMODE with Token Ring

With OpenVMS AXP Version 6.1 using Token Ring only, the IO$M_SHOW_MAP qualifier, when used with IO$_SENSEMODE, returns the current setting of the mapping table. The P2 buffer is filled with the current multicast to functional address mapping information. The entries are 16 bytes long and are in the format shown in Figure 4–13. This QIO requires PHY_IO privilege.

Figure 4–13 Format of IO$M_SHOW_MAP P2 Buffer



ZK–6793A–GE

The multicast address and functional address mask are returned in canonical format (that is, not bit-reversed). The following errors may occur:

- SS$_BUFFEROVF—The passed buffer is not large enough to hold all the data required for the operation.
- SS$_BADPARAM—Not able to get read access to buffer or zero length buffer passed.

For more information, see the *OpenVMS I/O User's Reference Manual*.

4.25.11.3 IO$M_SHOW_ROUTE Functional Modifier to IO$_SENSEMODE with Token Ring

With OpenVMS AXP Version 6.1 using Token Ring only, the IO$M_SHOW_ROUTE qualifier, when used with IO$_SENSEMODE, returns the current value of the source routing cache table. Each entry is 64 bytes long.

Figure 4–14 shows the format of the returned P2 buffer.

Figure 4–14 Format of IO$M_SHOW_ROUTE P2 Buffer

ZK–6794A–GE

Table 4–38 shows possible states of the entry.

Table 4–38 States of the Entry

Value	Name	Description
0	LOCAL	Address is reachable on the attached ring.
1	STALE	Entry is stale (inactive).
2	UNKNOWN	Route to the address is unknown.
3	DELETED	Entry is marked for deletion.
4	KNOWN	Route is known and the route is stored in the Routing Information String.
5	EXPLORING	Route to the address is currently being explored.

The LAN address is returned in canonical format (that is, not bit-reversed). The timers are recorded as seconds before expiration. The transmit and receive timers are initialized from the NMA$C_PCLI_A_TIM parameter, the discovery timer is initialized from the NMA$C_PCLI_ROUTEDIS parameter, and the stale timer is initialized to 10 minutes (600 seconds). The following errors may occur:

- SS$_BUFFEROVF—The passed buffer is not large enough to hold all the data required for the operation.
- SS$_BADPARAM—Not able to get read access to buffer or zero length buffer passed.

For more information, see the *OpenVMS I/O User's Reference Manual*.

5

DECwindows Motif Display Server New Features

The DECwindows Motif display server included with the OpenVMS AXP Version 6.1 operating system includes several new features. This chapter describes those features.

For more information about customizing and configuring your display server, see *Managing DECwindows Motif for OpenVMS Systems* in the DECwindows Motif Version 1.2 for OpenVMS documentation set.

5.1 DECwindows Server Customization Parameters

In OpenVMS AXP Version 6.1 systems, several new symbols have been created that you can use to customize the DECwindows Motif display server. In previous versions of OpenVMS, these parameters could be defined using logical names.

These customizations modify the default behavior of the DECwindows display server. Table 5–1 lists the new symbols for Version 6.1, a short description of what each parameter controls, and the default value. The parameters are grouped in Table 5–1 in the following areas:

- Server process
- Transport/connection
- Font caching
- Keyboard
- Extensions
- Screen saver
- Mouse
- Backing store
- Error reporting

To customize the DECwindows Motif display server environment, perform the following steps:

1. Copy the template file SYS$COMMON:[SYSMGR]DECW$PRIVATE_SERVER_SETUP.TEMPLATE to SYS$COMMON:[SYSMGR]DECW$PRIVATE_SERVER_SETUP.COM.
2. Edit the command file and search for the section entitled *Cluster Common or Standalone Workstation Setup*.
3. Add any customizations in that section, as in the following example:

```
$ DECW$SERVER_WSQUOTA == "10000"
```

4. Restart the DECwindows server with the following command:

```
$ @SYS$STARTUP:DECW$STARTUP RESTART
```

Caution

Restarting the DECwindows server disconnects all current client processes.

For a complete list of DECwindows display server parameters, descriptions, and customization examples, see *Managing DECwindows Motif for OpenVMS Systems* in the DECwindows Motif Version 1.2 for OpenVMS documentation set.

Table 5–1 DECwindows Customization Parameters

Parameter	Description	Default
Server Process		
DECW$SERVER_WSDEF	Defines process limits in pages.	3000
DECW$SERVER_WSQUOTA	Defines maximum amount of physical memory pages.	6000
DECW$SERVER_WSEXTENT	Defines limit on physical memory pages.	WSMAX[1]
DECW$SERVER_PAGE_FILE	Defines the maximum amount of virtual memory.	50000
DECW$SERVER_FILE_LIMIT	Defines the maximum number of files the server can open at one time.	200
DECW$SERVER_ENQUEUE_LIMIT	Defines the maximum number of outstanding locks that are used in sharing resources, particularly files, between processes.	512
Transport Setup		
DECW$SERVER_QUANTUM	Defines the maximum number of milliseconds that the server spends processing requests for a client before switching to the next client.	64
DECW$XPORT_SYNC_TIMEOUT	Defines the transport timeout value (in milliseconds). If the client does not take action to empty its buffers before the timeout, the server disconnects the client.	30000
DECW$SERVER_RETRY_WRITE_TIMEOUT	Defines the number of milliseconds at which time the DECwindows server stops attempting transport write requests.	500
DECW$SERVER_RETRY_WRITE_INTERVAL	Defines the number of milliseconds between failed transport write-request attempts.	30000

[1]Equal to the WSMAX system parameter

(continued on next page)

Table 5–1 (Cont.) DECwindows Customization Parameters

Parameter	Description	Default
16-Bit Font Caching Setup		
DECW$SERVER_FONT_CACHE_SIZE	Defines the number of 16-bit characters that remain resident in server memory at any one time. If a character is required that is not in the cache, the least recently used entry is flushed to make room for the new character which is read from the appropriate font file on disk. The minimum value for DECW$SERVER_FONT_CACHE_SIZE is 1024. Setting the value to 0 disables this feature.	0
DECW$SERVER_FONT_CACHE_UNIT	Defines size of the memory cells used to store the characters (glyphs) in memory. The minimum value for DECW$SERVER_FONT_CACHE_UNIT is 32. It should be set large enough to contain most glyphs in any of the 16-bit fonts that are used on the system. Glyphs can span multiple memory cells.	128
Keyboard Setup		
DECW$SERVER_BELL_BASE_VOLUME	Determines the bell volume of the keyboard.	50
DECW$SERVER_KEYCLICK_VOLUME	Determines the key click volume of the keyboard.	0
DECW$SERVER_ENABLE_KB_AUTOREPEAT	Causes a character to repeat itself automatically while a key is pressed.	True
Extensions		
DECW$SERVER_EXTENSIONS	Defines which loadable extensions should be activated (For more information, see Section 5.1.1).	See Table 5–2.
DECW$SERVER_ENABLE_ACCESSX	Defines the default state of the AccessX extension.	False
Screen Saver Setup		
DECW$SERVER_SCREEN_SAVER_PREFER_BLANKING	Determines the method by which screen saver is performed. When the value is True (the default), the DECwindows server causes the video device driver to turn off the video signal when the screen saver timeout expires. When the value is False, the DECwindows server clears the screen when the timeout expires.	True
DECW$SERVER_SCREEN_SAVER_TIMEOUT	Defines the initial time (in seconds) before the screen saver is activated, after which the screen saver interval takes effect.	600

(continued on next page)

Table 5–1 (Cont.) DECwindows Customization Parameters

Parameter	Description	Default
Screen Saver Setup		
DECW$SERVER_SCREEN_SAVER_INTERVAL	Defines the number of seconds the server waits before repainting the screen background.	600
Mouse Setup		
DECW$SERVER_MOUSE_THRESHOLD	Defines the minimum motions of the mouse (in pixels) at which time the DECwindows server is notified of the motion.	4
Backing Store Setup		
DECW$SERVER_DEFAULT_BACKING_STORE	Enables backing store of windows in three types of window states: (0) Indeterminate, depends on the server and the device being used; (1) Enable backing store only when window is mapped; (2) Enable backing store at all times.	0
DECW$SERVER_DISABLE_SAVE_UNDER	Records window information that may be hidden when another window is placed on top.	False
Error Reporting Setup		
DECW$SERVER_DUMP	Adds the /DUMP qualifier to the DCL command RUN causing a process dump if the server crashes.	False
DECW$CLIENT_ERROR_THRESHOLD	Defines the number of protocol errors generated by a single client above which the client will be terminated.	1
DECW$SERVER_ERROR_THRESHOLD	Defines the total number of server errors allowed before generating a warning message.	10

5.1.1 Server Extensions

A feature of all the X display servers is the ability to support server extensions. These are additions to the base display server that interpret additional protocol requests and perform new or improved functions.

The OpenVMS AXP display server supports dynamically loaded extensions. These extensions are contained in separate shareable images and are activated when the display server is started.

Use the symbol DECW$SERVER_EXTENSIONS to identify which loadable server extensions to load at startup.

Table 5–2 lists the available server extensions.

Table 5–2 Loadable Server Extensions

Extension Name	Description
Adobe-DPS-Extension[1]	Display PostScript
D2DX	Digital 2D performance enhancement
DEC-XTRAP[1]	Event trapping and simulation
Multi-Buffering[1]	Multi-buffered windows for smooth animation
Xie[1]	X Imaging Extension–performs imaging operations locally

[1]Preloaded by default

Table 5–3 lists the built-in server extensions.

Table 5–3 Built-in Server Extensions

Extension Name	Description
AccessX	Keyboard enhancements for people with movement impairments
DEC-Server-Mgmt-Extension	Server management functions used exclusively by the session manager
MIT-SHM	Shared memory fast PutImage support
MIT-SUNDRY-NONSTANDARD	Miscellaneous bug compatibility mode control
SHAPE	Nonrectangular windows
XTEST	Simple event trapping and simulation

5.2 Shared Memory Extension Support

Shared memory extension support provides the capability to share memory between XImages. This is essentially a version of the XImage interface where the actual image data is stored in a shared memory segment (global section), which is accessible by both the client and server. Consequently, the image does not need to be moved through the Xlib interprocess communication channel. For large images, use of this facility can result in dramatic performance increases.

For more information about using shared memory extensions, see the *DECwindows Motif Version 1.2 for OpenVMS Release Notes*.

Note

OpenVMS DECwindows Motif Version 1.2 for AXP is required to use this extension. Version 1.2 provides the client routines needed to interface with this extension in the SYS$SHARE:DECW$EXTLIBSHR.EXE shareable image.

5.3 Using LAT Transport

The DECwindows Motif AXP display server supports LAT as a network transport mechanism for displaying to OpenVMS Alpha AXP workstations.

Before you can use LAT transport, you must start the LAT software on both the DECwindows client and server systems. Incoming and outgoing messages must be enabled on both the client and server systems. Refer to the *OpenVMS System Manager's Manual* for more information about LAT transport. For information about how to use the LATCP utility, see the *OpenVMS System Management Utilities Reference Manual*.

The LAT service called "X$SERVER" must be present on the display server system in order to use LAT as a transport. Define the following logical to force the DECwindows startup procedure to create the LAT service.

```
$ DEFINE /SYSTEM DECW$INSTALL_XTERMINAL SERVER
```

You must define this logical before DECwindows is started. It is suggested that you define this logical in SYS$MANAGER:SYLOGICALS.COM.

To enable LAT as a transport for your display server, modify the file SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM to include the LAT transport in the list of possible transports. For example:

```
$ decw$server_transports == "LOCAL, DECNET, TCPIP, LAT"
```

On the client system, use the following procedure to to display your application using LAT transport:

1. Define the following logical in SYS$MANAGER:SYLOGICALS.COM

   ```
   $ DEFINE/SYSTEM DECW$INSTALL_XTERMINAL TRUE
   ```

2. Restart the DECwindows server with the following command:

   ```
   $ @SYS$STARTUP:DECW$STARTUP RESTART
   ```

3. Set the display with the following command:

   ```
   $ SET DISPLAY CREATE/TRANSPORT=LAT/NODE=node_name
   ```

5.4 AccessX Support

AccessX provides features to help people with different disabilities to interact with workstations running DECwindows Motif for OpenVMS AXP. These features can make it easier for people to use the the keyboard and the mouse. Users interact with workstations as usual, by entering commands and manipulating menus and dialog boxes. However, with AccessX features, they have an easier time performing these input operations.

To select and customize the AccessX features, execute the AccessX Configuration utility by entering the following command:

```
$ SET DEFAULT DECW$EXAMPLES:
$ RUN ACCESSX
```

For more information about using the AccessX Configuration utility, see the *DECwindows Motif Version 1.2 for OpenVMS Release Notes*.

5.5 New Keymap Name Support

This section lists the new DECwindows Motif keymap names in the display server. These keymap definition files should be used for both the PCXAL model keyboard as well as the LK443 and LK444 keyboards. Table 5–4 is arranged based on the language for which each keyboard is designed and includes keyboard model numbers. The *xx* designation indicates either LK or PC.

For all other keymap names and information about changing the default keyboard layout, see *Managing DECwindows Motif for OpenVMS Systems*.

Table 5–4 New Keymap Names

Language	Model	DECwindows Keymap Name
Austrian/German	LK444-(AG)	AUSTRIAN_GERMAN_LK444AG_*xx*
	PCXAL-(AG,FG,GG)	
Belgian/French	LK444-(AP)	BELGIAN_FRENCH_LK444AP_*xx*
	PCXAL-(AP,FP,GP)	
British	LK444-(AE)	BRITISH_LK444AE_*xx*
	PCXAL-(AE,FE,GE)	
Canadian/French	LK444-(AC,CC)	CANADIAN_FRENCH_LK444AC_*xx*
	PCXAL-(AC,GC)	
Czech	LK401-(BV)	CZECH_LK401_BV
Danish	LK444-(AD)	DANISH_LK444AD_*xx*
	PCXAL-(AD,FD,GD)	
Dutch	LK444-(CH)	DUTCH_LK444AH_*xx*
	PCXAL-(CH)	
Finnish	LK444-(CA)	FINNISH_LK444CA_*xx*
	PCXAL-(CA,JA)	
Flemish	LK444-(AB)	FLEMISH_LK444AB_*xx*
	PCXAL-(AB,FB,GB)	
Hungarian	LK401-(BQ)	HUNGARIAN_LK401_BQ
Italian	LK444-(AI)	ITALIAN_LK444AI_*xx*
	PCXAL-(AI,FI,GI)	
North American	PCXAL-(GQ)	NORTH_AMERICAN_LK443AA_*xx*
Norwegian	LK444-(AN)	NORWEGIAN_LK444AN_*xx*
	PCXAL-(AN,FN,GN)	
Polish	LK401-(BP)	POLISH_LK401_BP
Portuguese	LK444-(AV)	PORTUGUESE_LK444AV_*xx*
	PCXAL-(AV,FV,GV)	
Russian	LK401-(BT)	RUSSIAN_LK401_BT
Slovak	LK401-(CZ)	SLOVAK_LK401_CZ
Spanish	LK444-(AS)	SPANISH_LK444AS_*xx*
	PCXAL-(AS,FS,GS)	

(continued on next page)

Table 5–4 (Cont.) New Keymap Names

Language	Model	DECwindows Keymap Name
Swedish	LK444-(CA)	SWEDISH_LK444CA_*xx*
	PCXAL-(CA,JA)	
Swiss/French	PCXAL-(FK,GK)	SWISS_FRENCH_LK444AK_*xx*
Swiss/German	PCXAL-(GL)	SWISS_GERMAN_LK444CH_*xx*
US	LK443-(AA)	US_LK443AA_*xx*
	PCXAL-(FA,GA)	

5.5.1 Keymaps of LK Version Keyboard

The *_LK versions of these keymap files define the function, cursor control, and keypad keys to match those available on an LK401 style keyboard. Table 5–5 lists the key and what is reported when that key is pressed.

Table 5–5 Key Mapping for LK Model Keyboards

Key	Reported as:
Insert	Find
Home	Insert Here
Page Up	Remove
Delete	Select
End	Prev
Page Down	Next
Num Lock	PF1
/	PF2
*	PF3
-	PF4
keypad +	keypad ,
Caps Lock keypad +	keypad -
Print Screen	Help
Scroll Lock	Do

The numeric and the period key (.) keypad keys always report their numeric values.

5.5.2 Keymaps of PC Version Keyboard

When the *_PC versions of the keymap files are selected, the special keys are treated according to the labels on the keys. Some OpenVMS applications have been written assuming that the keyboard layout is LK401 style. If you select a *_PC version, it may be difficult to interact with these applications.

If a *_PC keymap is selected and the Num Lock key is pressed, the Num Lock indicator light will be lit. Pressing Num Lock a second time will turn the indicator light off.

Note

DECwindows Motif Version 1.2 for OpenVMS is required to report the numeric value for a numeric key without having to press the Shift key while the Num Lock light is on.

5.5.3 Simulating LK401 Function Keys

On systems which support PC style keyboards, use the following method to simulate LK401 keys which are not available on PC style keyboards:

1. Hold down the Caps Lock key
2. Press any of the F1 through F10 keys to simulate the F11 through F20 keys

5.6 New Qualifier for the Font Compiler

The DECwindows X11 display server includes a font compiler that enables programmers to convert an ASCII bitmap distribution format (BDF) into binary portable compiled format (PCF). The server uses a PCF file to display a font.

To invoke the font compiler, use the following format:

```
FONT filespec [ /OUTPUT[=filename] [ /[NO]INK_METRICS ]
```

The filespec parameter specifies the BDF file to be converted to PCF format. A file name is required. The default value of the optional file type is .BDF.

The /OUTPUT qualifier specifies the file name and file type of the resulting PCF file. The default output file name is the file name of the BDF file being converted. The default output PCF file type is .PCF.

PCF output files consist of a header file that contains font information, character metrics, and the image of each character in the font.

The new qualifier /NOINK_METRICS causes the font compiler to omit the font's ink metrics information (per character). This will reduce overall font in-memory usage. The /INK_METRICS qualifier is used by default. Ink metrics define the smallest box that includes the entire symbol, without any extraneous space outside the symbol.

5.7 Font Server Support

The DECwindows X11 display server includes a font server renderer that supports the use of font servers to access fonts. This allows font files to be distributed on systems other than the system on which the display server is running.

You can add font servers to the font path by defining the symbol DECW$FONT_SERVERS in the site-specific server setup file SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM. Depending on the transport type, use one of the following:

- To add support for the TCP transport, specify TCP as the network connection type that will communicate with the font server. Replace *node* with the Internet node name. Specify the Transmission Control Protocol\Internet

Protocol (TCP/IP) port number for *port_number*, as shown in the following example:

```
$ DECW$FONT_SERVERS == "TCP/node::port_number"
```

- To add support for the DECnet transport, specify DECNET as the network connection type that will communicate with the font server. Replace *node* with the DECnet node name. Specify the DECnet object name for *network_object* as in the following example:

```
$ DECW$FONT_SERVERS == "DECNET/node::network_object"
```

Multiple font servers can be added by defining the symbol as a comma-separated list. The symbol is not case sensitive.

Note

A font server is not provided with OpenVMS AXP, only access to existing font servers. The port number or network object are defined when a font server is configured.

5.8 New Font Support

This section lists the new font names that are included in the display server. The following table indicates the font type and the table in which you can find the specific font name and file name:

Table	Font Type
Table 5–6	Lucida Bright
Table 5–7	Lucida Sans
Table 5–8	Lucida Typewriter
Table 5–9	MIT Miscellaneous
Table 5–10	Type 1 Scalable
Table 5–11	VT330

See the *VMS DECwindows Guide to Xlib Programming: MIT C Binding* for all other fonts included in the display server.

Table 5–6 Lucida Bright Font Names

.PCF File Name	Font Name
LUCIDABRIGHT08	-B&H-LucidaBright-Medium-R-Normal--8-80-75-75-P-45-ISO8859-1
LUCIDABRIGHT10	-B&H-LucidaBright-Medium-R-Normal--10-100-75-75-P-56-ISO8859-1
LUCIDABRIGHT12	-B&H-LucidaBright-Medium-R-Normal--12-120-75-75-P-68-ISO8859-1
LUCIDABRIGHT14	-B&H-LucidaBright-Medium-R-Normal--14-140-75-75-P-80-ISO8859-1
LUCIDABRIGHT18	-B&H-LucidaBright-Medium-R-Normal--18-180-75-75-P-103-ISO8859-1
LUCIDABRIGHT19	-B&H-LucidaBright-Medium-R-Normal--19-190-75-75-P-109-ISO8859-1
LUCIDABRIGHT24	-B&H-LucidaBright-Medium-R-Normal--24-240-75-75-P-137-ISO8859-1

(continued on next page)

Table 5–6 (Cont.) Lucida Bright Font Names

.PCF File Name	Font Name
LUCIDABRIGHT_DEMI08	-B&H-LucidaBright-DemiBold-R-Normal--8-80-75-75-P-47-ISO8859-1
LUCIDABRIGHT_DEMI10	-B&H-LucidaBright-DemiBold-R-Normal--10-100-75-75-P-59-ISO8859-1
LUCIDABRIGHT_DEMI12	-B&H-LucidaBright-DemiBold-R-Normal--12-120-75-75-P-71-ISO8859-1
LUCIDABRIGHT_DEMI14	-B&H-LucidaBright-DemiBold-R-Normal--14-140-75-75-P-84-ISO8859-1
LUCIDABRIGHT_DEMI18	-B&H-LucidaBright-DemiBold-R-Normal--18-180-75-75-P-107-ISO8859-1
LUCIDABRIGHT_DEMI19	-B&H-LucidaBright-DemiBold-R-Normal--19-190-75-75-P-114-ISO8859-1
LUCIDABRIGHT_DEMI24	-B&H-LucidaBright-DemiBold-R-Normal--24-240-75-75-P-143-ISO8859-1
LUCIDABRIGHT_ DEMIITALIC08	-B&H-LucidaBright-DemiBold-I-Normal--8-80-75-75-P-48-ISO8859-1
LUCIDABRIGHT_ DEMIITALIC10	-B&H-LucidaBright-DemiBold-I-Normal--10-100-75-75-P-59-ISO8859-1
LUCIDABRIGHT_ DEMIITALIC12	-B&H-LucidaBright-DemiBold-I-Normal--12-120-75-75-P-72-ISO8859-1
LUCIDABRIGHT_ DEMIITALIC14	-B&H-LucidaBright-DemiBold-I-Normal--14-140-75-75-P-84-ISO8859-1
LUCIDABRIGHT_ DEMIITALIC18	-B&H-LucidaBright-DemiBold-I-Normal--18-180-75-75-P-107-ISO8859-1
LUCIDABRIGHT_ DEMIITALIC19	-B&H-LucidaBright-DemiBold-I-Normal--19-190-75-75-P-114-ISO8859-1
LUCIDABRIGHT_ DEMIITALIC24	-B&H-LucidaBright-DemiBold-I-Normal--24-240-75-75-P-143-ISO8859-1
LUCIDABRIGHT_ITALIC08	-B&H-LucidaBright-Medium-I-Normal--8-80-75-75-P-45-ISO8859-1
LUCIDABRIGHT_ITALIC10	-B&H-LucidaBright-Medium-I-Normal--10-100-75-75-P-57-ISO8859-1
LUCIDABRIGHT_ITALIC12	-B&H-LucidaBright-Medium-I-Normal--12-120-75-75-P-67-ISO8859-1
LUCIDABRIGHT_ITALIC14	-B&H-LucidaBright-Medium-I-Normal--14-140-75-75-P-80-ISO8859-1
LUCIDABRIGHT_ITALIC18	-B&H-LucidaBright-Medium-I-Normal--18-180-75-75-P-102-ISO8859-1
LUCIDABRIGHT_ITALIC19	-B&H-LucidaBright-Medium-I-Normal--19-190-75-75-P-109-ISO8859-1
LUCIDABRIGHT_ITALIC24	-B&H-LucidaBright-Medium-I-Normal--24-240-75-75-P-136-ISO8859-1
LUCIDABRIGHT08_100DPI	-B&H-LucidaBright-Medium-R-Normal--11-80-100-100-P-63-ISO8859-1
LUCIDABRIGHT10_100DPI	-B&H-LucidaBright-Medium-R-Normal--14-100-100-100-P-80-ISO8859-1
LUCIDABRIGHT12_100DPI	-B&H-LucidaBright-Medium-R-Normal--17-120-100-100-P-96-ISO8859-1
LUCIDABRIGHT14_100DPI	-B&H-LucidaBright-Medium-R-Normal--20-140-100-100-P-114-ISO8859-1
LUCIDABRIGHT18_100DPI	-B&H-LucidaBright-Medium-R-Normal--25-180-100-100-P-142-ISO8859-1
LUCIDABRIGHT19_100DPI	-B&H-LucidaBright-Medium-R-Normal--26-190-100-100-P-149-ISO8859-1
LUCIDABRIGHT24_100DPI	-B&H-LucidaBright-Medium-R-Normal--34-240-100-100-P-193-ISO8859-1
LUCIDABRIGHT_DEMI08_ 100DPI	-B&H-LucidaBright-DemiBold-R-Normal--11-80-100-100-P-66-ISO8859-1
LUCIDABRIGHT_DEMI10_ 100DPI	-B&H-LucidaBright-DemiBold-R-Normal--14-100-100-100-P-84-ISO8859-1
LUCIDABRIGHT_DEMI12_ 100DPI	-B&H-LucidaBright-DemiBold-R-Normal--17-120-100-100-P-101-ISO8859-1

(continued on next page)

Table 5–6 (Cont.) Lucida Bright Font Names

.PCF File Name	Font Name
LUCIDABRIGHT_DEMI14_100DPI	-B&H-LucidaBright-DemiBold-R-Normal--20-140-100-100-P-118-ISO8859-1
LUCIDABRIGHT_DEMI18_100DPI	-B&H-LucidaBright-DemiBold-R-Normal--25-180-100-100-P-149-ISO8859-1
LUCIDABRIGHT_DEMI19_100DPI	-B&H-LucidaBright-DemiBold-R-Normal--26-190-100-100-P-155-ISO8859-1
LUCIDABRIGHT_DEMI24_100DPI	-B&H-LucidaBright-DemiBold-R-Normal--34-240-100-100-P-202-ISO8859-1
LUCIDABRIGHT_DEMIITALIC08_100DPI	-B&H-LucidaBright-DemiBold-I-Normal--11-80-100-100-P-66-ISO8859-1
LUCIDABRIGHT_DEMIITALIC10_100DPI	-B&H-LucidaBright-DemiBold-I-Normal--14-100-100-100-P-84-ISO8859-1
LUCIDABRIGHT_DEMIITALIC12_100DPI	-B&H-LucidaBright-DemiBold-I-Normal--17-120-100-100-P-101-ISO8859-1
LUCIDABRIGHT_DEMIITALIC14_100DPI	-B&H-LucidaBright-DemiBold-I-Normal--20-140-100-100-P-119-ISO8859-1
LUCIDABRIGHT_DEMIITALIC18_100DPI	-B&H-LucidaBright-DemiBold-I-Normal--25-180-100-100-P-149-ISO8859-1
LUCIDABRIGHT_DEMIITALIC19_100DPI	-B&H-LucidaBright-DemiBold-I-Normal--26-190-100-100-P-156-ISO8859-1
LUCIDABRIGHT_DEMIITALIC24_100DPI	-B&H-LucidaBright-DemiBold-I-Normal--34-240-100-100-P-203-ISO8859-1
LUCIDABRIGHT_ITALIC08_100DPI	-B&H-LucidaBright-Medium-I-Normal--11-80-100-100-P-63-ISO8859-1
LUCIDABRIGHT_ITALIC10_100DPI	-B&H-LucidaBright-Medium-I-Normal--14-100-100-100-P-80-ISO8859-1
LUCIDABRIGHT_ITALIC12_100DPI	-B&H-LucidaBright-Medium-I-Normal--17-120-100-100-P-96-ISO8859-1
LUCIDABRIGHT_ITALIC14_100DPI	-B&H-LucidaBright-Medium-I-Normal--20-140-100-100-P-113-ISO8859-1
LUCIDABRIGHT_ITALIC18_100DPI	-B&H-LucidaBright-Medium-I-Normal--25-180-100-100-P-142-ISO8859-1
LUCIDABRIGHT_ITALIC19_100DPI	-B&H-LucidaBright-Medium-I-Normal--26-190-100-100-P-148-ISO8859-1
LUCIDABRIGHT_ITALIC24_100DPI	-B&H-LucidaBright-Medium-I-Normal--34-240-100-100-P-194-ISO8859-1

Table 5–7 Lucida Sans Font Names

.PCF File Name	Font Name
LUCIDA_BOLDITALICSANS08	-B&H-Lucida-Bold-I-Normal-Sans-8-80-75-75-P-49-ISO8859-1
LUCIDA_BOLDITALICSANS10	-B&H-Lucida-Bold-I-Normal-Sans-10-100-75-75-P-67-ISO8859-1
LUCIDA_BOLDITALICSANS12	-B&H-Lucida-Bold-I-Normal-Sans-12-120-75-75-P-79-ISO8859-1

(continued on next page)

Table 5–7 (Cont.) Lucida Sans Font Names

.PCF File Name	Font Name
LUCIDA_ BOLDITALICSANS14	-B&H-Lucida-Bold-I-Normal-Sans-14-140-75-75-P-92-ISO8859-1
LUCIDA_ BOLDITALICSANS18	-B&H-Lucida-Bold-I-Normal-Sans-18-180-75-75-P-119-ISO8859-1
LUCIDA_ BOLDITALICSANS19	-B&H-Lucida-Bold-I-Normal-Sans-19-190-75-75-P-122-ISO8859-1
LUCIDA_ BOLDITALICSANS24	-B&H-Lucida-Bold-I-Normal-Sans-24-240-75-75-P-151-ISO8859-1
LUCIDA_BOLDSANS08	-B&H-Lucida-Bold-R-Normal-Sans-8-80-75-75-P-50-ISO8859-1
LUCIDA_BOLDSANS10	-B&H-Lucida-Bold-R-Normal-Sans-10-100-75-75-P-66-ISO8859-1
LUCIDA_BOLDSANS12	-B&H-Lucida-Bold-R-Normal-Sans-12-120-75-75-P-79-ISO8859-1
LUCIDA_BOLDSANS14	-B&H-Lucida-Bold-R-Normal-Sans-14-140-75-75-P-92-ISO8859-1
LUCIDA_BOLDSANS18	-B&H-Lucida-Bold-R-Normal-Sans-18-180-75-75-P-120-ISO8859-1
LUCIDA_BOLDSANS19	-B&H-Lucida-Bold-R-Normal-Sans-19-190-75-75-P-122-ISO8859-1
LUCIDA_BOLDSANS24	-B&H-Lucida-Bold-R-Normal-Sans-24-240-75-75-P-152-ISO8859-1
LUCIDA_ITALICSANS08	-B&H-Lucida-Medium-I-Normal-Sans-8-80-75-75-P-45-ISO8859-1
LUCIDA_ITALICSANS10	-B&H-Lucida-Medium-I-Normal-Sans-10-100-75-75-P-59-ISO8859-1
LUCIDA_ITALICSANS12	-B&H-Lucida-Medium-I-Normal-Sans-12-120-75-75-P-71-ISO8859-1
LUCIDA_ITALICSANS14	-B&H-Lucida-Medium-I-Normal-Sans-14-140-75-75-P-82-ISO8859-1
LUCIDA_ITALICSANS18	-B&H-Lucida-Medium-I-Normal-Sans-18-180-75-75-P-105-ISO8859-1
LUCIDA_ITALICSANS19	-B&H-Lucida-Medium-I-Normal-Sans-19-190-75-75-P-108-ISO8859-1
LUCIDA_ITALICSANS24	-B&H-Lucida-Medium-I-Normal-Sans-24-240-75-75-P-136-ISO8859-1
LUCIDA_SANS08	-B&H-Lucida-Medium-R-Normal-Sans-8-80-75-75-P-45-ISO8859-1
LUCIDA_SANS10	-B&H-Lucida-Medium-R-Normal-Sans-10-100-75-75-P-58-ISO8859-1
LUCIDA_SANS12	-B&H-Lucida-Medium-R-Normal-Sans-12-120-75-75-P-71-ISO8859-1
LUCIDA_SANS14	-B&H-Lucida-Medium-R-Normal-Sans-14-140-75-75-P-81-ISO8859-1
LUCIDA_SANS18	-B&H-Lucida-Medium-R-Normal-Sans-18-180-75-75-P-106-ISO8859-1
LUCIDA_SANS19	-B&H-Lucida-Medium-R-Normal-Sans-19-190-75-75-P-108-ISO8859-1
LUCIDA_SANS24	-B&H-Lucida-Medium-R-Normal-Sans-24-240-75-75-P-136-ISO8859-1
LUCIDA_ BOLDITALICSANS08_100DPI	-B&H-Lucida-Bold-I-Normal-Sans-11-80-100-100-P-69-ISO8859-1
LUCIDA_ BOLDITALICSANS10_100DPI	-B&H-Lucida-Bold-I-Normal-Sans-14-100-100-100-P-90-ISO8859-1
LUCIDA_ BOLDITALICSANS12_100DPI	-B&H-Lucida-Bold-I-Normal-Sans-17-120-100-100-P-108-ISO8859-1
LUCIDA_ BOLDITALICSANS14_100DPI	-B&H-Lucida-Bold-I-Normal-Sans-20-140-100-100-P-127-ISO8859-1
LUCIDA_ BOLDITALICSANS18_100DPI	-B&H-Lucida-Bold-I-Normal-Sans-25-180-100-100-P-159-ISO8859-1
LUCIDA_ BOLDITALICSANS19_100DPI	-B&H-Lucida-Bold-I-Normal-Sans-26-190-100-100-P-166-ISO8859-1

(continued on next page)

Table 5–7 (Cont.) Lucida Sans Font Names

.PCF File Name	Font Name
LUCIDA_BOLDITALICSANS24_100DPI	-B&H-Lucida-Bold-I-Normal-Sans-34-240-100-100-P-215-ISO8859-1
LUCIDA_BOLDSANS08_100DPI	-B&H-Lucida-Bold-R-Normal-Sans-11-80-100-100-P-70-ISO8859-1
LUCIDA_BOLDSANS10_100DPI	-B&H-Lucida-Bold-R-Normal-Sans-14-100-100-100-P-89-ISO8859-1
LUCIDA_BOLDSANS12_100DPI	-B&H-Lucida-Bold-R-Normal-Sans-17-120-100-100-P-108-ISO8859-1
LUCIDA_BOLDSANS14_100DPI	-B&H-Lucida-Bold-R-Normal-Sans-20-140-100-100-P-127-ISO8859-1
LUCIDA_BOLDSANS18_100DPI	-B&H-Lucida-Bold-R-Normal-Sans-25-180-100-100-P-158-ISO8859-1
LUCIDA_BOLDSANS19_100DPI	-B&H-Lucida-Bold-R-Normal-Sans-26-190-100-100-P-166-ISO8859-1
LUCIDA_BOLDSANS24_100DPI	-B&H-Lucida-Bold-R-Normal-Sans-34-240-100-100-P-216-ISO8859-1
LUCIDA_ITALICSANS08_100DPI	-B&H-Lucida-Medium-I-Normal-Sans-11-80-100-100-P-62-ISO8859-1
LUCIDA_ITALICSANS10_100DPI	-B&H-Lucida-Medium-I-Normal-Sans-14-100-100-100-P-80-ISO8859-1
LUCIDA_ITALICSANS12_100DPI	-B&H-Lucida-Medium-I-Normal-Sans-17-120-100-100-P-97-ISO8859-1
LUCIDA_ITALICSANS14_100DPI	-B&H-Lucida-Medium-I-Normal-Sans-20-140-100-100-P-114-ISO8859-1
LUCIDA_ITALICSANS18_100DPI	-B&H-Lucida-Medium-I-Normal-Sans-25-180-100-100-P-141-ISO8859-1
LUCIDA_ITALICSANS19_100DPI	-B&H-Lucida-Medium-I-Normal-Sans-26-190-100-100-P-147-ISO8859-1
LUCIDA_ITALICSANS24_100DPI	-B&H-Lucida-Medium-I-Normal-Sans-34-240-100-100-P-192-ISO8859-1
LUCIDA_SANS08_100DPI	-B&H-Lucida-Medium-R-Normal-Sans-11-80-100-100-P-63-ISO8859-1
LUCIDA_SANS10_100DPI	-B&H-Lucida-Medium-R-Normal-Sans-14-100-100-100-P-80-ISO8859-1
LUCIDA_SANS12_100DPI	-B&H-Lucida-Medium-R-Normal-Sans-17-120-100-100-P-96-ISO8859-1
LUCIDA_SANS14_100DPI	-B&H-Lucida-Medium-R-Normal-Sans-20-140-100-100-P-114-ISO8859-1
LUCIDA_SANS18_100DPI	-B&H-Lucida-Medium-R-Normal-Sans-25-180-100-100-P-142-ISO8859-1
LUCIDA_SANS19_100DPI	-B&H-Lucida-Medium-R-Normal-Sans-26-190-100-100-P-147-ISO8859-1
LUCIDA_SANS24_100DPI	-B&H-Lucida-Medium-R-Normal-Sans-34-240-100-100-P-191-ISO8859-1

Table 5–8 Lucida Typewriter Font Names

.PCF File Name	Font Name
LUCIDATYPEWRITER_BOLDSANS08	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-8-80-75-75-M-50-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS10	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-10-100-75-75-M-60-ISO8859-1

(continued on next page)

Table 5–8 (Cont.) Lucida Typewriter Font Names

.PCF File Name	Font Name
LUCIDATYPEWRITER_BOLDSANS12	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-12-120-75-75-M-70-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS14	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-14-140-75-75-M-90-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS18	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-18-180-75-75-M-110-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS19	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-19-190-75-75-M-110-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS24	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-24-240-75-75-M-140-ISO8859-1
LUCIDATYPEWRITER_SANS08	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-8-80-75-75-M-50-ISO8859-1
LUCIDATYPEWRITER_SANS10	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-10-100-75-75-M-60-ISO8859-1
LUCIDATYPEWRITER_SANS12	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-12-120-75-75-M-70-ISO8859-1
LUCIDATYPEWRITER_SANS14	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-14-140-75-75-M-90-ISO8859-1
LUCIDATYPEWRITER_SANS18	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-18-180-75-75-M-110-ISO8859-1
LUCIDATYPEWRITER_SANS19	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-19-190-75-75-M-110-ISO8859-1
LUCIDATYPEWRITER_SANS24	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-24-240-75-75-M-140-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS08_100DPI	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-11-80-100-100-M-70-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS10_100DPI	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-14-100-100-100-M-80-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS12_100DPI	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-17-120-100-100-M-100-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS14_100DPI	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-20-140-100-100-M-120-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS18_100DPI	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-25-180-100-100-M-150-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS19_100DPI	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-26-190-100-100-M-159-ISO8859-1
LUCIDATYPEWRITER_BOLDSANS24_100DPI	-B&H-LucidaTypewriter-Bold-R-Normal-Sans-34-240-100-100-M-200-ISO8859-1
LUCIDATYPEWRITER_SANS08_100DPI	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-11-80-100-100-M-70-ISO8859-1
LUCIDATYPEWRITER_SANS10_100DPI	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-14-100-100-100-M-80-ISO8859-1
LUCIDATYPEWRITER_SANS12_100DPI	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-17-120-100-100-M-100-ISO8859-1
LUCIDATYPEWRITER_SANS14_100DPI	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-20-140-100-100-M-120-ISO8859-1

(continued on next page)

Table 5–8 (Cont.) Lucida Typewriter Font Names

.PCF File Name	Font Name
LUCIDATYPEWRITER_SANS18_100DPI	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-25-180-100-100-M-150-ISO8859-1
LUCIDATYPEWRITER_SANS19_100DPI	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-26-190-100-100-M-159-ISO8859-1
LUCIDATYPEWRITER_SANS24_100DPI	-B&H-LucidaTypewriter-Medium-R-Normal-Sans-34-240-100-100-M-200-ISO8859-1

Table 5–9 MIT Miscellaneous Font Names

.PCF File Name	Font Name
5x7	-misc-fixed-medium-r-normal--7-70-75-75-c-50-iso8859-1
7x14b	-misc-fixed-bold-r-normal--14-130-75-75-c-70-iso8859-1
7x14rk	-misc-fixed-medium-r-normal--14-130-75-75-c-70-jisx0201.1976-0
8x16rk	-sony-fixed-medium-r-normal--16-120-100-100-c-80-jisx0201.1976-0
12x24rk	-sony-fixed-medium-r-normal--24-170-100-100-c-120-jisx0201.1976-0
hanglg16	-daewoo-gothic-medium-r-normal--16-120-100-100-c-160-ksc5601.1987-0
hanglm16	-daewoo-mincho-medium-r-normal--16-120-100-100-c-160-ksc5601.1987-0
hanglm24	-daewoo-mincho-medium-r-normal--24-170-100-100-c-240-ksc5601.1987-0
heb6x13	-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-8
heb8x13	-misc-fixed-medium-r-normal--13-120-75-75-c-80-iso8859-8
jiskan16	-jis-fixed-medium-r-normal--16-150-75-75-c-160-jisx0208.1983-0
jiskan24	-jis-fixed-medium-r-normal--24-230-75-75-c-240-jisx0208.1983-0
k14	-misc-fixed-medium-r-normal--14-130-75-75-c-140-jisx0208.1983-0
nil2	-misc-nil-medium-r-normal--2-20-75-75-c-10-misc-fontspecific

Table 5–10 Type 1 Scalable

.PFA File Name	Font Name
UTRG___	-adobe-utopia-medium-r-normal--0-0-0-0-p-0-iso8859-1
UTI___	-adobe-utopia-medium-i-normal--0-0-0-0-p-0-iso8859-1
UTB___	-adobe-utopia-bold-r-normal--0-0-0-0-p-0-iso8859-1
UTBI___	-adobe-utopia-bold-i-normal--0-0-0-0-p-0-iso8859-1

Table 5–11 VT330 Font Names

.PCF File Name	Font Name
VT33018	-DEC-VT330-Medium-R-Normal--20-180-75-75-C-100-ISO8859-1
VT33036	-DEC-VT330-Medium-R-Normal--40-360-75-75-C-200-ISO8859-1
VT330_BOLD18	-DEC-VT330-Bold-R-Normal--20-180-75-75-C-100-ISO8859-1
VT330_BOLD36	-DEC-VT330-Bold-R-Normal--40-360-75-75-C-200-ISO8859-1

(continued on next page)

Table 5–11 (Cont.) VT330 Font Names

.PCF File Name	Font Name
VT330_BOLD_DBLWIDE18	-DEC-VT330-Bold-R-Double Wide--20-180-75-75-C-200-ISO8859-1
VT330_BOLD_DBLWIDE_ DECTECH18	-DEC-VT330-Bold-R-Double Wide--20-180-75-75-C-200-DEC-DECtech
VT330_BOLD_DECTECH18	-DEC-VT330-Bold-R-Normal--20-180-75-75-C-100-DEC-DECtech
VT330_BOLD_DECTECH36	-DEC-VT330-Bold-R-Normal--40-360-75-75-C-200-DEC-DECtech
VT330_BOLD_NARROW18	-DEC-VT330-Bold-R-Narrow--20-180-75-75-C-60-ISO8859-1
VT330_BOLD_NARROW36	-DEC-VT330-Bold-R-Narrow--40-360-75-75-C-120-ISO8859-1
VT330_BOLD_NARROW_ DECTECH18	-DEC-VT330-Bold-R-Narrow--20-180-75-75-C-60-DEC-DECtech
VT330_BOLD_NARROW_ DECTECH36	-DEC-VT330-Bold-R-Narrow--40-360-75-75-C-120-DEC-DECtech
VT330_BOLD_WIDE18	-DEC-VT330-Bold-R-Wide--20-180-75-75-C-120-ISO8859-1
VT330_BOLD_WIDE_ DECTECH18	-DEC-VT330-Bold-R-Wide--20-180-75-75-C-120-DEC-DECtech
VT330_DBLWIDE18	-DEC-VT330-Medium-R-Double Wide--20-180-75-75-C-200-ISO8859-1
VT330_DBLWIDE_ DECTECH18	-DEC-VT330-Medium-R-Double Wide--20-180-75-75-C-200-DEC-DECtech
VT330_DECTECH18	-DEC-VT330-Medium-R-Normal--20-180-75-75-C-100-DEC-DECtech
VT330_DECTECH36	-DEC-VT330-Medium-R-Normal--40-360-75-75-C-200-DEC-DECtech
VT330_NARROW18	-DEC-VT330-Medium-R-Narrow--20-180-75-75-C-60-ISO8859-1
VT330_NARROW36	-DEC-VT330-Medium-R-Narrow--40-360-75-75-C-120-ISO8859-1
VT330_NARROW_ DECTECH18	-DEC-VT330-Medium-R-Narrow--20-180-75-75-C-60-DEC-DECtech
VT330_NARROW_ DECTECH36	-DEC-VT330-Medium-R-Narrow--40-360-75-75-C-120-DEC-DECtech
VT330_WIDE18	-DEC-VT330-Medium-R-Wide--20-180-75-75-C-120-ISO8859-1
VT330_WIDE_DECTECH18	-DEC-VT330-Medium-R-Wide--20-180-75-75-C-120-DEC-DECtech

6
System Management Examples

This chapter describes examples of tools that the system manager might find useful in performing a number of common system management tasks:

Description	Reference
Displaying job process information	Section 6.1
Terminating processes	Section 6.2
Submitting periodic batch jobs	Section 6.3
Displaying system roots	Section 6.4
Deleting a directory or its contents, or both	Section 6.5

"More examples" has been a frequent request of OpenVMS customers. The system management examples described in this chapter represent a first step in responding to customers' requests.

It is the intention of the OpenVMS documentation team to continue to increase the number and variety of examples we provide. If you have examples you would be willing to share with us, or suggestions of examples you would like to see included in future releases, please contact us via one of the following:

- Internet electronic mail: OPENVMSDOC@ZKO.MTS.DEC.COM
- Fax 603-881-0120 Attn: OpenVMS Documentation, ZK03-4/U08
- Digital Equipment Corporation
  Information Design and Consulting
  OpenVMS Documentation
  110 Spit Brook Road, ZK03-4/U08
  Nashua, NH 03062-2698
  USA

Copies of the examples in this chapter are available on the OpenVMS AXP distribution media in the [DOCUMENTATION.V061.EXAMPLES] directory. The first example listed above must be installed using VMSINSTAL; the rest of the examples are DCL command procedures. For more information on how to access the files containing these tools, see the *OpenVMS AXP CD–ROM User's Guide*.

The examples provided here are not intended to represent the only way or the best way to accomplish tasks for each user. Rather, they are examples that some OpenVMS users have found to be useful.

You should examine the purpose and logic of each example to determine which ones are suitable for your environment. For examples in this section where the source code is available on line, you might want to modify the code to meet your site's specific needs.

Note

Digital does not warrant that any System Management Examples provided herein will meet customers' requirements, operate without interruption, or be error free.

6.1 Job Process Information (JPI) Utility

JPI is a wide area network tool that you can use to create a one-line-per-process display of information. Depending on the criteria you select, you can display information about all or specific processes on local nodes or clusters and on remote nodes or clusters.

Although JPI can extract information from remote nodes and remote VMSclusters, this feature is available only on nodes that are running JPI Version 2.0 or higher and that have the JPI DECnet object properly defined. (Defining the JPI DECnet object is part of the installation procedure.)

JPI displays the following values for each process selected:

BASEPRIORITY	NODENAME
CURRENT STATE	PID
CURRENT WORKING SET SIZE	PROCESS
IMAGENAME	TERMINAL
MODE	USERNAME

For a description of each value in this list, refer to $GETJPI in the *OpenVMS System Services Reference Manual*.

Example 6–1 shows a sample JPI display using the /SELECTION qualifier, which displays the current default selection criteria.

Example 6–1 JPI Display

```
                    Job/Process Information from ALPHA/VMS Version 6.1

                 From node DRPCTX (DEC 3000 Model 400) at  9-MAR-1995 11:28:02.48

PID      Term   Username      Process Name  Image Name  PriB Work State M Node
------------------------------------------------------------------------------
00000085         SYSTEM...... IPCACP....... IPCACP......  8  2048 HIB   O DRPCTX
00000086         SYSTEM...... ERRFMT....... ERRFMT......  7  3488 HIB   O DRPCTX
00000087         SYSTEM...... OPCOM........ OPCOM.......  6  1088 HIB   O DRPCTX
00000088         AUDIT$SERVER AUDIT_SERVER. AUDIT_SERVER  8  3488 HIB   O DRPCTX
00000089         SYSTEM...... JOB_CONTROL.. JBC$JOB_CONT  8  2048 HIB   O DRPCTX
0000008A         SYSTEM...... QUEUE_MANAGER QMAN$QUEUE_M  8  4448 HIB   O DRPCTX
0000008B         SYSTEM...... TP_SERVER.... TPSERV......  8  2048 HIB   O DRPCTX
0000008C         DECNET...... NETACP....... NETACP......  8  3488 HIB   O DRPCTX
0000008D         DECNET...... EVL.......... EVL.........  4  3488 HIB   N DRPCTX
0000008E         SYSTEM...... REMACP....... REMACP......  8  1088 HIB   O DRPCTX
00000090         SYSTEM...... DECW$SERVER_0 DECW$SERVER_  6 19792 HIB   O DRPCTX
00000091         SYSTEM...... DECW$SESSION. DECW$SESSION  4 18800 LEF   I DRPCTX
00000093 MBA25:. SYSTEM...... DECW$MWM..... DECW$MWM....  4 11600 LEF   I DRPCTX
00000094 MBA27:. SYSTEM...... SYSTEM....... DECW$CLOCK..  4 11600 LEF   I DRPCTX
00000095         SYSTEM...... VUE$SYSTEM_3. ............  4  2000 LEF   I DRPCTX
00000096         SYSTEM...... VUE$SYSTEM_4. ............  4  2000 LEF   I DRPCTX
00000097 MBA42:. SYSTEM...... DECW$TE_0097. DECW$TERMINA  5 14000 LEF   O DRPCTX
00000098 FTA5:.. SYSTEM...... _FTA5:....... ............  4  2000 LEF   I DRPCTX
00000099 FTA6:.. SYSTEM...... _FTA6:....... ............  4  2000 LEF   I DRPCTX
0000009A RTA1:.. SMACK....... SMACK........ ............  4  2000 LEF   I DRPCTX
0000009B RTA2:.. SMACK....... _RTA2:....... JPI.........  4  4400 CUR   I DRPCTX

Total of 21 Processes using 116912 pages (58 mb) of physical memory

Command Line:  JPI/SELECTION/OUTPUT=SELECTION.TXT

Defaults:      /NODENAME=DRPCTX
               /USERNAME=*
               /IMAGENAME=*
               /MODE=(INTERACTIVE,BATCH,NETWORK,OTHER)
               /JOBTYPE=(LOCAL,DIALUP,REMOTE,BATCH,NETWORK,DETACHED)
               /NOSORT
               /PROCESS=*
               /TERMINAL=*
               /HW_NAME=*
               /BASEPRIORITY=(MINIMUM=0,MAXIMUM=16)
               /HEADER
               /TRAILER
               /NOSWAP
               /NOSYMBOLS
               /PRIVILEGE=ANY
```

The following defaults shown in Example 6–1 are input/output options and are listed in Table 6–2: /NOSORT, /HEADER, /TRAILER, /NOSWAP, and /NOSYMBOLS. The rest of the defaults are selection criteria, which are listed in Table 6–1.

Privileges Required

For you to access information for processes within your group, your calling process must have the GROUP system privilege. For you to view processes outside your group, your process must have the WORLD system privilege. JPI displays only processes for which you have required privileges.

6.1.1 Installing JPI

You must install JPI using VMSINSTAL. You can install onto a system running VMS Version 5.3 or higher and OpenVMS AXP Version 6.1 or higher.

To install JPI, copy the save set JPI021.A from the distribution media CD–ROM to your system. Then run VMSINSTAL specifying the JPI021.A save set. (See Example 6–2.)

For more information about how to access the files on the distribution media, see the *OpenVMS AXP CD–ROM User's Guide*.

Files that the installation kit modifies or creates are the following:

File	Description	Created or Updated
SYS$LIBRARY:DCLTABLES.EXE	DCL tables executable image	Updated
SYS$SYSTEM:JPI.EXE	JPI utility executable image	Created
SYS$HELP:HELPLIB.HLB	JPI Help file	Updated
SYS$MANAGER:JPI021.RELEASE_NOTES	JPI release notes	Created
JPIBAS.BAS	BASIC sub-module	Created (optionally)
JPIGBLSYM.MAR	Global symbol definitions	Created (optionally)
JPICLD.CLD	JPI command definitions	Created (optionally)
JPI.COB	COBOL code for JPI	Created (optionally)

Caution

This utility uses the DCLTABLES CLI for a command named JPI. JPI is *not* a Digital registered command and therefore may, in the future, conflict with a Digital registered DCL command.

Example 6–2 contains a sample log file for an installation of JPI using VMSINSTAL.

Note

During the installation, if you answer "Yes" to the question "Do you want the source files?", be sure to specify a directory that already exists.

Example 6–2 Sample VMSINSTAL Log File for JPI Utility

```
Username: TMACK
Password:

   Welcome to OpenVMS AXP (TM) Operating System, Version 6.1 on node DRPCTX
    Last interactive login on Friday, 11-MAR-1995 16:25:42.72
    Last non-interactive login on Friday, 11-MAR-1995 16:00:17.98

$ @sys$update:vmsinstal jpi021.a sys$login options i

         OpenVMS AXP Software Product Installation Procedure Version 6.1

%VMSINSTAL-W-NOTSYSTEM, You are not logged in to the SYSTEM account.
%VMSINSTAL-W-ACTIVE, The following processes are still active:
 TMACK

The following products will be processed:
  JPI V2.1

        Beginning installation of JPI 2.1 at 16:26

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.

       The following source files are available:
        1) jpi.cob        - The main program
        2) jpicld.cld     - called by jpi.cob when jpi invoked remotely
        3) jpibas.bas     - called by jpi.cob for remote access file I/O
        4) jpigblsym.mar  - global symbol definitions

* Do you want the source files?  [NO]? yes

* Enter Directory for Source(s)  [SYS$SYSDEVICE:[TMACK]]:

       This version of JPI has the capability to be called from a remote
       JPI image on another DECnet node. In order for this capability to
       be utilized, a DECnet object must be defined on this node/cluster.

* Do you want to define the JPI DECnet object?  [YES]? yes

* Do you want to run the IVP after the installation [YES]? yes

* Do you want to purge files replaced by this installation [YES]? yes

  No further questions will be asked.

  The network object JPI has been defined.  If you wish
  to view this, run SYS$SYSTEM:NCP and issue the command
        NCP> LIST OBJECT JPI CHAR

%JPI-I-LINK, Linking the jpi image ...

------------------------------------------------------------------------
POST INSTALLATION INSTRUCTIONS

For cluster installations:

1) You must execute the DCL command INSTALL/REPLACE
   SYS$LIBRARY:DCLTABLES.EXE on each node of the VMScluster that
   utilizes this system disk. This has been done on this node by
   the installation procedure. You can use the SYSMAN utility to
   accomplish this by using the following example:

   MCR SYSMAN
   SYSMAN> SET ENVIRONMENT/CLUSTER
   SYSMAN> DO INSTALL REPLACE SYS$LIBRARY:DCLTABLES.EXE
   SYSMAN> EXIT
```

(continued on next page)

Example 6–2 (Cont.) Sample VMSINSTAL Log File for JPI Utility

```
2) On each node of this cluster that uses the same DECnet object
   database as this node, you must issue an appropriate network
   object definition and set command.

   Examples:

   Phase IV:
      $ MCR NCP SET OBJ JPI ALL
   Phase V: (DECNET/OSI)
      $ mcr ncl
      NCL> CREATE NODE 0 SESSION CONTROL APPLICATION JPI
      NCL> SET NODE 0 SESSION CONTROL APPLICATION JPI ADDRESSES = {NAME=JPI} -
      ,CLIENT =  -
      ,INCOMING ALIAS = TRUE -
      ,INCOMING PROXY = TRUE -
      ,OUTGOING ALIAS = FALSE -
      ,OUTGOING PROXY = TRUE -
      ,NODE SYNONYM = TRUE -
      ,IMAGE NAME = SYS$SYSTEM:JPI.EXE -
      ,INCOMING OSI TSEL =

   This step is required is lieu of restarting the entire
   DECnet and must be done only once.

   For ALL installations:

   For each user who is currently logged in, either issue the following
   DCL command SET COMMAND/TABLE=SYS$LIBRARY:DCLTABLES, or have the user
   log out and then log in again in order to receive the new DCL command
   for JPI.

-------------------------------------------------------------------------

%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...
Now Executing the IVP for JPI V2.1

IVPSUCCESS-I-FNDEXE, JPI V2.1 image located in sys$system.
Invoking JPI ...

   .
   .
   .

IVPSUCCESS-I-JPIOK, Invocation of JPI command produced expected results.
IVPSUCCESS-I-FNDHLP Utility JPI help module located in SYS$HELP:HELPLIB.
IVPSUCCESS-I-IVPCMPLT, IVP has completed successfully.

 Installation of JPI 2.1 completed at 16:27

    Adding history entry in VMI$ROOT:[SYSUPD]VMSINSTAL.HISTORY

    Creating installation data file: VMI$ROOT:[SYSUPD]JPI021.VMI_DATA

 VMSINSTAL procedure done at 16:27
$ log
  TMACK        logged out at 11-MAR-1995 16:27:51.71
```

6.1.2 Using JPI

When you use JPI, you can choose combinations of the JPI selection criteria shown in Table 6–1 and the input/output options shown in Table 6–2.

Table 6–1 describes selection criteria that you can use to indicate which processes you want to include in the display. The selection criteria apply to the scope of the JPI process search, whether the search is confined to a local node, a local cluster, a remote node, a remote cluster, or any combination of these.

Table 6–1 JPI Selection Criteria

Option	Description
BASE-PRIORITY	**Format:** JPI/BASEPRIORITY=(keyword[,...]) Allows you to select processes based on the base priority of the process. If you specify multiple keywords, separate them with commas and enclose the list in parentheses. Following are explanations of the keywords MINIMUM and MAXIMUM: • MAXIMUM=n Selects processes that have a base priority equal to or less than the specified parameter. The default for MAXIMUM is 16. • MINIMUM=n Selects processes that have a base priority equal to or greater than the specified parameter. The default for MINIMUM is 0. • (MINIMUM=n,MAXIMUM=n) Selects processes that have a base priority within the parameters specified. To specify a single base priority, specify MINIMUM and MAXIMUM as the same value. The default is all priorities. Examples: 1. JPI/BASEPRIORITY=(MINIMUM=4,MAXIMUM=10) The command in this example selects only processes with base priorities between 4 and 10. 2. JPI/BASEPRIORITY=MINIMUM=9 The command in this example selects processes with base priorities between 9 and 16. 3. JPI/BASEPRIORITY=(MINIMUM=4,MAXIMUM=4) The command in this example selects processes with a base priority of 4.
HW_NAME	**Format:** JPI/HW_NAME=(cpu_hardware_name[,...]) Displays processes running on the specified CPU. The CPU name is the same value as that returned from the $GETSYI("HW_NAME") lexical function and the HW_TYPE item on a $ SHOW CLUSTER listing. You can use the wildcard characters % and * and up to 10 arguments within the specification. If you specify multiple CPU hardware names, separate them with commas and enclose the list in parentheses. If a CPU hardware name includes one or more spaces, tabs, or the slash (/) character, you must use quotation marks around the entire name; for example, "*/GPX". You must use /HW_NAME with either the /NODENAME or the /CLUSTER qualifier. The default is all CPU types (that is, /HW_NAME=*). **Example:** JPI/HW_NAME=("VAX 6000*","*VLC")/CLUSTER The command in this example selects all processes running on any CPU on the local cluster either beginning with the value VAX 6000 or having VLC in the CPU hardware name.

(continued on next page)

Table 6–1 (Cont.) JPI Selection Criteria

Option	Description
IMAGENAME	**Format:** JPI/IMAGENAME=(imagename[,...]) Specifies the image name or names to be used as selection criteria. When you use this qualifier, JPI issues an implied /SWAP. (See the description of the /SWAP qualifier in Table 6–2.) You can use wildcard characters in image names. If you specify multiple image names, separate them with commas and enclose the list in parentheses. If you use the /IMAGE qualifier, an implied /SWAP is issued to JPI to ensure that all process image names are available for review. You can override this implied /SWAP with a /NOSWAP qualifier. The default is all image names (that is, /IMAGENAME=*). **Example:** JPI/IMAGENAME=(*SMB*,DOC*) The command in this example selects all processes that have image names containing SMB and all image names that start with DOC. **Performance note:** Because the $PROCESS_SCAN system service does not support an image name selection, JPI performs this selection after the $GETJPI system service has been called. As a result, IMAGENAME is a costly option in terms of performance; that is, you will notice a degradation in display response time when you use this qualifier.
JOBTYPE	**Format:** JPI/JOBTYPE=(jobtype[,...]) Specifies the process job type or types to be used as selection criteria. Possible job types are LOCAL, DIALUP, REMOTE, BATCH, NETWORK, and DETACHED. You need to enter only enough text to uniquely identify the job type. If you specify multiple job types, separate them with commas and enclose the list in parentheses. The default is all job types (that is, /JOBTYPE=ALL). **Example:** JPI/JOBTYPE=(LOCAL,BATCH) The command in this example selects processes that have the LOCAL and BATCH job type attributes.
MODE	**Format:** JPI/MODE=(mode[,...]) Specifies the process mode or modes to be used as selection criteria. The modes are INTERACTIVE, BATCH, NETWORK and OTHER. You need to enter only enough text to uniquely identify the mode. If you specify multiple modes, separate them with commas and enclose the list in parentheses. The default is all modes (that is, /MODE=ALL). **Example:** JPI/MODE=(INTERACTIVE,BATCH) The command in this example displays INTERACTIVE and BATCH process modes.

(continued on next page)

Table 6–1 (Cont.) JPI Selection Criteria

Option	Description
NODENAME	**Format:** JPI/NODENAME[=(nodename[,...])] Specifies the cluster node or nodes to be used as selection criteria. You can use DCL wildcard characters in node names. If you specify multiple node names, separate them with commas and enclose the list in parentheses. JPI ignores the /NODENAME qualifier when you specify it on a non-clustered system. You cannot use /NODENAME with the /CLUSTER qualifier. To retrieve information from the local node when you use the /RNODENAME or /RCLUSTER qualifier, you must also use the /NODENAME qualifier. Node name values can be logical names. Only first-level logical name translation is performed on logical names; that is, a specified value cannot translate to a logical name that, in turn, translates to another logical name. You can specify up to 50 node names. If a node name is a logical name that points to multiple values, each of these corresponding values counts against the limit of 50. **Example:** $ DEFINE CI SNAK,NOLAND,IPM31 $ JPI/NODENAME=(CI,PCDE*) The JPI command following the definition actually uses up 4 of the possible 50 node name values—three for the translated values from the logical CI, and one for the wildcarded explicit value of PCDE*. If you do not specify a node name, the default is the current node. **Example:** JPI/NODENAME=(HICCUP,DSWK%) The command in this example selects the following: • All processes on the node HICCUP • Processes on nodes with node names that are 5 characters long and have DSWK as the first 4 characters
PRIVILEGE	**Format:** JPI/PRIVILEGE=(privilege[,...]) Specifies the process privileges to be used as selection criteria. You cannot use wildcards. If you specify multiple privileges, separate them with commas and enclose the list in parentheses. You can concatenate privilege names with the plus (+) sign if you place quotation marks around concatenated privileges; for example, "SYSNAM+READALL". The default is all image names (that is, /PRIVILEGE=ANY).

(continued on next page)

Table 6–1 (Cont.) JPI Selection Criteria

Option	Description
	Example: JPI/PRIVILEGE=("READALL+GRPPRV",SYSNAM,BYPASS,SYSPRV) The command in this example finds a process match if the process currently has READALL *and* GRPPRV privileges, SYSNAM privilege, BYPASS privilege, or SYSPRV privilege. Privileges available to an authorized process on a VMS Version 6.1 system are the following: ACNT, ALLSPOOL, ALTPRI, AUDIT, BUGCHK, BYPASS, CMEXEC, CMKRNL, DETACH, DIAGNOSE, DOWNGRADE, EXQUOTA, GROUP, GRPNAM, GRPPRV, IMPORT, LOG_IO, MOUNT, NETMBX, OPER, PFNMAP, PHY_IO, PRMCEB, PRMGBL, PRMMBX, PSWAPM, READALL, SECURITY, SETPRV, SHARE, SHMEM, SYSGBL, SYSLCK, SYSNAM, SYSPRV, TMPMBX, UPGRADE, VOLPRO, WORLD Many privileges, with varying degrees of power and potential system impact, are available. See the *OpenVMS System Manager's Manual* for a more detailed discussion of available privileges. **Performance note:** Using the /PRIVILEGE qualifier can adversely affect the performance of JPI.
PROCESS	**Format:** JPI/PROCESS=(process[,...]) Specifies the process name or names to be used as selection criteria. You can use wildcard characters in process names. If you specify multiple process names, separate them with commas and enclose the list in parentheses. The default is all process names (that is, /PROCESS=*). **Example:** JPI/PROCESS=(SYSTEM,JA*) The command in this example selects all processes with process names of SYSTEM and process names that begin with the characters JA.
RNODENAME	**Format:** JPI/RNODENAME=(nodename[,...]) Specifies the remote node name or names for which the selection criteria apply. As with the /NODENAME qualifier, these values can be logical names. The limit for node name values is 50. The JPI DECnet object must be properly defined on the remote node; this object must point to a valid JPI image (Version 2.0 or higher). You can pass access control information as part of the node name specification. In the absence of optional proxy access information on the target node or nodes, default DECnet access information applies. Specifying /RNODENAME negates default selection criteria on the issuing node. If you also want process information about the issuing node or cluster, you must specify a /NODENAME or /CLUSTER qualifier on the command line. **Example:** JPI/RNODENAME=(IM4U"rburns greatscotts",COOKIE::) The command in this example issues remote calls to nodes IM4U and COOKIE: • For node IM4U, JPI will utilize the user name *rburns* and password *greatscotts* for logging in to the JPI process. • To log in to node COOKIE, JPI must have proxy information defined for the calling *username:password*.

(continued on next page)

Table 6–1 (Cont.) JPI Selection Criteria

Option	Description
TERMINAL /NOTERMINAL	**Formats:** JPI/TERMINAL=(terminal_name[,...]) JPI/TERMINAL=none JPI/TERMINAL=any Specifies the process-associated terminal name or names to be used as selection criteria. JPI selects a process if any of the supplied values match the name of a terminal associated with a process. You can enter up to 10 separate terminal specifications. If you specify multiple terminal names, separate them with commas and enclose the list in parentheses. You can use wildcards anywhere within the arguments. When you specify /TERMINAL=none or /NOTERMINAL, JPI selects only processes that have no associated terminal. This is equivalent to /TERMINAL=" ". The default is all terminals (that is, /TERMINAL=*). Examples: 1. JPI/TERMINAL=(RTA*,FTA*) The command in this example selects processes with associated terminal names that begin with RTA and FTA. 2. JPI/TERMINAL=(none,RTA*) The command in this example selects processes with no associated terminal and terminal names that begin with the characters RTA. 3. JPI/NOTERMINAL The command in this example selects only those processes that have no associated terminals. 4. JPI/TERMINAL=ANY The command in this example selects processes that have a non-blank terminal name associated with them. Specify /TERMINAL=ANY if you want to specify processes that have a terminal associated with them. The ANY value deselects processes that have no terminal association.
USERNAME	**Format:** JPI/USERNAME=(username[,...]) Specifies the user name or names to be used as selection criteria. You can use wildcard characters in user names. If you specify multiple user names, separate them with commas and enclose the list in parentheses. The default is all user names (that is, /USERNAME=*). **Example:** JPI/USERNAME=(KAHILL,JA*) The command in this example selects all processes that have the user name KAHILL and all processes with user names beginning with the characters JA.

You can sort the selection criteria in Table 6–1 in any combination of field order, in either ascending or descending order.

Example: JPI/SORT=(WSSIZE=DESCENDING,PROCESS)

Table 6–2 shows the input and output options that are available with JPI.

Table 6–2 JPI Input/Output Options

Option	Type of Option	Description
CLUSTER	Input	Causes JPI to search all cluster nodes for processes matching the input selection criteria. The CLUSTER option is ignored when specified from a noncluster system. You cannot negate this qualifier (it is equal to /NODENAME=*), nor can you specify this qualifier with the /NODENAME qualifier. The default is /NOCLUSTER.
HEADER	Output	Controls whether a header line is displayed. You cannot specify this qualifier with the /SUMMARY qualifier. The default is /HEADER.
OUTPUT	Output	**Format:** /OUTPUT=filename Controls where the output of the command is sent. You must have write access to the specified area. The default area for a file is the current device and directory.
RCLUSTER	Input	**Format:** /RCLUSTER=(clustername[,...]) Specifies the remote cluster or clusters for which the selection criteria apply. If you specify multiple cluster names, separate them with commas and enclose the list in parentheses. JPI will make selections on all nodes currently executing in the cluster and meeting all other specified selection criteria. As with the /NODENAME qualifier, these values can be logical names. The limit for cluster name values is 50. You must define the JPI DECnet object on the remote node; this object must point to a valid JPI image (Version 2.0 or higher). You can pass access control information as part of the node name specification. In the absence of optional proxy access information on the target node or nodes, default DECnet access information applies. Specifying /RCLUSTER negates default selection criteria on the issuing node. If you also want cluster information about the issuing node or cluster, you must explicitly specify a /NODENAME or /CLUSTER qualifier on the command line.
SELECTION	Output	Indicates that selection criteria are to be displayed. The command line and any default values will be displayed. (See Example 6–1.) If you use the /SELECTION qualifier and also supply password information with an /RCLUSTER qualifier, an /RNODENAME qualifier, or both, the literal word "password" is substituted for the actual password on the displayed selection line. The default is /NOSELECTION.

(continued on next page)

Table 6–2 (Cont.) JPI Input/Output Options

Option	Type of Option	Description
SORT	Output	**Format:** SORT=(keyword[,...]) Specifies the sort sequence for the output. Valid keywords are BASEPRIORITY, IMAGE, JOBTYPE, MODE, NODENAME, PID, PROCESS, STATE, TERMINAL, USERNAME, and WSSIZE. If you specify multiple keywords, separate them with commas and enclose the list in parentheses. You cannot use this qualifier with the /SUMMARY qualifier. The default is /NOSORT. If you specify /SORT without any keywords, the default sort sequence is by ascending user name. The default sorting order is ascending. You can specify the order for each keyword specified. **Example:** /SORT=(USERNAME=DESCENDING,WSSIZE) The command in this example sorts the output first by descending user name, and then, within user name, by ascending working set size.
SUMMARY /NOSUMMARY	Output	Limits the display to only a summary line. You cannot specify this qualifier with the /HEADER or /SORT qualifiers. The default is /SUMMARY.
SWAP	Input	Causes the $GETJPI system service to swap in any processes that have been swapped out. You can use this option to retrieve any process information in the outswapped process data structures. Currently, the process image name and working set size are maintained in swappable data structures; you can retrieve them by specifying the /SWAP qualifier. The default is /NOSWAP. **Performance note:** Inswapping can cause considerable degradation in JPI performance.
SYMBOLS /NOSYMBOLS	Output	Instructs JPI to create local string symbols that equate to the formatted detail lines. Symbol names are of the form JPI$DETAIL_LINE_nnnn, where *nnnn* is a zero-filled and right-justified integer. In addition, the local string symbol JPI$ITEM_COUNT contains an integer value representing the total number of processes matching the selection criteria. When there are no matches, the symbol is equated to 0.
TRAILER /NOTRAILER	Output	Causes a one-line summary of the number of processes to be displayed after the last detail process line has been displayed. These processes are ones found within the selection criteria to which you had access privileges. The default is /TRAILER.

Usage Note

Select remote node information by using the /RNODENAME qualifier; select remote cluster information by specifying the /RCLUSTER qualifier; or you can use both qualifiers together. The syntax for these qualifiers is identical to that specified with the /NODENAME qualifier. (See Table 6–1.)

In addition, you can specify access control information in the standard DECnet syntax for each of the remote nodes or clusters you specify.

6.2 Terminating Processes

The KILLPROCESS.COM procedure allows you to terminate processes on the local node or cluster.

You can use the procedure to terminate only processes for which you have privilege.

To invoke the procedure, enter the following command:

```
$ @KILLPROCESS.COM
```

The system returns the process identification (PID), terminal identifier, user name, image name, process name, and node for each process on the system.

```
    Proc Id  Term    Username     Image Name      Process Name         Node

 1) 27C058B4 FTA558: SMITH....... VMSHELP........ _FTA558:............ BHAK..
 2) 27C014CE RTA10:. JONES....... ............... - Joe - ........... BHAK..
 3) 27C05EDC VTA771: WALTERS..... ............... _VTA771:............ BHAK..
 4) 27C034E8 VTA773: ADAMS....... RTPAD.......... _VTA773:............ BHAK..
 5) 27C06AF1 ....... DAVIS....... ............... Dan Davis........... BHAK..
 6) 27C060F2 RTA25:. JEFFRIES.... ............... _RTA25:............. BHAK..
 7) 27C04D27 FTA542: SMITH....... ............... SMITH............... BHAK..
 8) 26602A7A FTA145: SMITH....... ............... SMITH............... DELAND
 9) 26602AAC FTA154: SMITH....... ............... _FTA154:............ DELAND
10) 27803033 FTA111: SMITH....... ............... _FTA111:............ IPL31.
11) 2780309B FTA114: SMITH....... ............... SMITH............... IPL31.
12) 278002FD RTA12:. JONES....... ............... - Joe - ........... IPL31.
13) 212000C3 MBA182: JONES....... DECW$MWM....... DECW$MWM............ WRTR07
14) 212000C8 ....... JONES....... ............... VUE$JONES_4......... WRTR07
15) 212000D1 MBA197: JONES....... DECW$CLOCK..... VUE$JONES_5......... WRTR07
16) 212000D4 ....... JONES....... ............... VUE$JONES_6......... WRTR07
17) 212000DA FTA21:. JONES....... RTPAD.......... _FTA21:............. WRTR07
18) 2120026E FTA22:. JONES....... RTPAD.......... JONES............... WRTR07
19) 212000B0 FTA15:. JONES....... DECW$SESSION... - Joe - ........... WRTR07

Select Process to Terminate ('/' to Select Specific Users) [Exit]
```

You can choose which processes to terminate by selecting the number. To view all of the processes of one user, enter a slash (/). The system returns the following prompt:

```
Enter Username to Select From [*]
```

At this prompt, the default, the asterisk (*), displays the entire list of users again. However, if you enter SMITH, for example, the system returns a sorted list of Smith's processes from which you can then choose.

```
    Proc Id  Term    Username     Image Name      Process Name         Node

 1) 27C058B4 FTA558: SMITH....... VMSHELP........ _FTA558:............ BHAK..
 2) 27C04D27 FTA542: SMITH....... ............... SMITH............... BHAK..
 3) 26602A7A FTA145: SMITH....... ............... SMITH............... DELAND
 4) 26602AAC FTA154: SMITH....... ............... _FTA154:............ DELAND
 5) 27803033 FTA111: SMITH....... ............... _FTA111:............ IPL31.
 6) 2780309B FTA114: SMITH....... ............... SMITH............... IPL31.
```

When you have selected a process to terminate, the system will ask for confirmation by returning the following query:

```
Please Confirm! Kill this Process? [No]
```

If you choose no, the default, the system will return the complete list of processes. If you type yes, the system will terminate the process you selected.

For each process you terminate, user Smith will receive the following terminal message, where SYSTEM is the user name of the process terminating Smith's process:

```
Reply received on BHAK from user SYSTEM at _BHAK$RTA25:   16:32:17

Sorry.
Your Process was killed at  3-MAR-1994 16:32:17.46
By User SYSTEM at terminal RTA25: on Node BHAK
Terminal _BHAK$RTA25: has been notified.
```

You might use the procedure to terminate SMITH processes directly by entering the following command:

```
$ @KILLPROCESS.COM SMITH
```

In this example, the system returns the PID, terminal identifier, user name, image name, process name, and node for each process with the user name Smith.

```
      Proc Id  Term    Username     Image Name      Process Name         Node

 (1)  27C058B4 FTA558: SMITH....... VMSHELP........ _FTA558:............ BHAK..
 (2)  27C05EDC VTA771: SMITH....... ............... _VTA771:............ BHAK..
 (3)  27C034E8 VTA773: SMITH....... RTPAD.......... _VTA773:............ BHAK..
 (4)  27C06AF1 ....... SMITH....... ............... Sue Smith........... BHAK..
 (5)  27C060F2 RTA25:. SMITH....... ............... _RTA25:............. BHAK..
 (6)  27C04D27 FTA542: SMITH....... ............... SMITH.............. BHAK..
 (7)  26602A7A FTA145: SMITH....... ............... SMITH.............. DELAND
 (8)  26602AAC FTA154: SMITH....... ............... _FTA154:............ DELAND
 (9)  27803033 FTA111: SMITH....... ............... _FTA111:............ IPL31.
(10)  2780309B FTA114: SMITH....... ............... SMITH.............. IPL31.

Select Process to Terminate ('/' to Select Specific Users) [Exit])
```

You can then select any or all of the SMITH processes for termination.

For each process you terminate, user Smith will receive the same terminal message shown above.

You can also use wildcards to sort the list of processes by user. For example, you can obtain a listing of all processes with user names beginning with A and B by entering the following command:

```
$ @KILLPROCESS.COM A* B*
```

A copy of KILLPROCESS.COM is available on the OpenVMS AXP distribution media in the [DOCUMENTATION.V061.EXAMPLES] directory. For more information about how to access the files on the distribution media, see the *OpenVMS AXP CD–ROM User's Guide*.

6.3 Submitting Periodic Batch Jobs

The PERIODIC.TEMPLATE file can be used to create procedures that allow you to execute periodic, repetitive batch jobs. The procedure allows you to submit jobs 7 days a week, but is limited to executing on the hour.

For example, you may want to receive periodic reports of users on a system. To do this, first copy PERIODIC.TEMPLATE to a file named USERS.COM (or some other meaningful name).

Edit USERS.COM and modify the schedule table. For example, if you want the job to execute every business day at 06:00, 12:00, 18:00, and 00:00, the table should appear as follows:

```
$!
$!                              000000000011111111112222
$!                              012345678901234567890123
$       schedule_sunday    = "                         "
$       schedule_monday    = "1     1     1     1     "
$       schedule_tuesday   = "1     1     1     1     "
$       schedule_wednesday = "1     1     1     1     "
$       schedule_thursday  = "1     1     1     1     "
$       schedule_friday    = "1     1     1     1     "
$       schedule_saturday  = "                         "
```

To submit the procedure, enter the following command:

```
$ @USERS.COM "SHOW USERS/FULL"
```

Note that this command procedure can be used to issue DCL commands, as illustrated above, or to call another command procedure. To call another command procedure, enter the following, where PERIODIC.COM is the procedure containing the schedule and UPDATE.COM is the procedure to be submitted:

```
$ @PERIODIC.COM "@UPDATE.COM"
```

The procedure will run indefinitely, returning every 6 hours a listing of users on the system. To stop the procedure, issue a SHOW ENTRY command for your batch queue. When you have determined the entry number of the procedure, issue the command STOP/ID=nnn, where *nnn* is the number of the entry to be stopped.

A copy of PERIODIC.TEMPLATE is available on the OpenVMS AXP distribution media in the [DOCUMENTATION.V061.EXAMPLES] directory. For more information about how to access the files on the distribution media, see the *OpenVMS AXP CD–ROM User's Guide*.

6.4 Displaying System Roots

The FINDROOTS.COM procedure allows you to find out which system roots are available on your system. Using this information, you can reuse unused system roots, or, using SYS$MANAGER:CLUSTER_CONFIG.COM, you can delete roots that you no longer need.

When invoked, FINDROOTS scans for system roots in SYS$SYSDEVICE and displays the root, node name, DECnet address, last boot date and time, and hardware type of each system root it finds. The information FINDROOTS displays for each root directory that it finds comes from the ALPHAVMSSYS.PAR system parameter file.

The procedure also allows you to create logical names for roots; it creates a logical name table with an entry for each logical name specification.

Example 6–3 shows a sample FINDROOTS display, with column labels added for easy identification.

Example 6–3 Sample FINDROOTS Display

```
Executing WORK111:[SMACK]FINDROOTS.COM;19 from node RACKUP at 15-MAR-1995 15:22:03.76

System Root       Node     DECnet     Last Boot Date and Time     Hardware Type
                           Node
-------------     ----     ------     ----------------------      -------------
[SYS0.SYSEXE]     ASL31    19.0850     7-FEB-1995 12:25:04.66     VAX 6000-240
[SYS10.SYSEXE]    FOSATO   19.0699     1-MAR-1995 16:29:25.98     VAXstation 3100/GPX
[SYS11.SYSEXE]    SUBWAY   19.0345    23-FEB-1995 13:30:36.54     VAXstation 3100/GPX
[SYS12.SYSEXE]    SAMMY    19.0348    11-MAR-1995 10:53:26.43     VAXstation 2000
[SYS13.SYSEXE]    CRUMBA   19.0473    16-FEB-1995 12:17:59.88     VAXstation 3600 Series
[SYS14.SYSEXE]    FLOP     63.0135    18-JAN-1995 08:22:41.43     VAXstation 3100
[SYS15.SYSEXE]    MEMBER   19.0855    24-FEB-1995 12:19:47.75     VAXstation 3100/GPX
[SYS16.SYSEXE]    SIGNUP   63.0868    18-JAN-1995 08:53:00.16     VAXstation 3100/GPX
[SYS17.SYSEXE]    WRITIT   63.0090    21-JAN-1995 10:15:02.57     VAXstation 3100
[SYS18.SYSEXE]    MATCHU   19.0188    18-JAN-1995 08:36:04.10     VAXstation 4000-VLC
[SYS19.SYSEXE]    CAKE     19.0839    10-FEB-1995 09:36:42.90     VAXstation 4000-VLC
[SYS1A.SYSEXE]    APPLES   63.0261    18-JAN-1995 08:09:51.97     VAXstation 4000-VLC
[SYS1B.SYSEXE]    ONEDER   63.0213     7-FEB-1995 15:18:25.99     VAXstation 4000-VLC
[SYS1C.SYSEXE]    ONE4U    19.0235    22-MAR-1993 16:29:10.65
[SYS1D.SYSEXE]    DANCER   19.0329    23-FEB-1995 16:53:25.43     VAXstation 3100/GPX
[SYS1E.SYSEXE]    MAX      63.0162     1-MAR-1995 09:43:07.45     VAXstation 4000-VLC
[SYS1F.SYSEXE]    MINNIE   63.0041    18-JAN-1995 08:09:01.50     VAXstation 3100
[SYS2.SYSEXE]     RACKUP   19.0868     7-FEB-1995 13:13:50.03     VAX 6000-620
[SYS20.SYSEXE]    EZROAD   63.0100     8-MAR-1995 08:46:13.24     VAXstation 3100
[SYS21.SYSEXE]    FUMIN    19.0166    12-JUN-1993 16:34:00.23
[SYS22.SYSEXE]    HAPPY    63.0916     7-SEP-1993 14:10:24.36
[SYS23.SYSEXE]    MAKEUP   63.0740    18-JAN-1995 11:51:03.48     VAXstation 3100/GPX
```

Notes on the display columns:

- The system root, node name, and DECnet address are always displayed.
- The last boot date and time are displayed only if you specify a file from which FINDROOT can retrieve a boot date and time. FINDROOT.COM explains a method—and includes commands—for doing this.
- The hardware type is displayed for nodes that are currently running in a VMScluster.

If a hardware type is not listed for a node, and if the last boot date is old, you can probably reuse the node name or delete the root. Examples in Example 6–3 are the ONE4U, FUMIN, and HAPPY nodes.

The FINDROOTS procedure assumes that all system roots are in SYS$SYSDEVICE on the system that executes this procedure. Therefore, in a dual-architecture VMScluster, you must execute this procedure on each system disk.

A copy of FINDROOTS.COM is available on the OpenVMS AXP distribution media in the [DOCUMENTATION.V061.EXAMPLES] directory. For more information about how to access the files on the distribution media, see the *OpenVMS AXP CD–ROM User's Guide*.

6.5 Deleting Directory Trees

The DELTREE.COM procedure allows you to delete *all* files in a specified directory, all files in that directory's subdirectories, and, optionally, the directory and its subdirectories.

Verification is required before this procedure will delete a directory.

Caution

If you have write access to the specified directory and files, this procedure automatically converts your write access to delete access. Therefore, setting the protection on the directory or file will not prevent the directory or file from being deleted.

To invoke the procedure, enter the following command, where USER.FILES is the directory you want to delete:

```
$ @DELTREE.COM [USER.FILES]
```

Note that the procedure will operate on the [.FILES] subdirectory of the USER directory, not the USER directory itself.

The system returns the following query to verify that you want to delete the files in [USER.FILES] directory:

```
Confirm. Delete directory tree SYS$SYSDEVICE:[USER.FILES] [N]
```

If you answer yes, the system returns the following query:

```
Confirm. Remove directory files too [Y]
```

If you answer yes, the system will delete all of the files in the [USER.FILES] directory, along with any files contained in subdirectories of the [.FILES] directory. Directory files will also be deleted.

If you answer no, the files in [.USER] and its subdirectories will be deleted, but the subdirectory structure will remain intact.

If your current default directory is the [.FILES] directory, enter the following command to delete the directory's contents and its subdirectories:

```
$ @DELTREE.COM [-]FILES
```

The system will return the same queries shown above to verify your request.

Example 6–4 shows the messages generated by DELTREE.COM. In this example, the user has chosen to delete the files in the [.FILES] directory and leave the subdirectories intact.

Example 6–4 Sample DELTREE Display

```
$ SHOW DEFAULT
  SYS$SYSDEVICE:[SMACK]

$ SET DEFAULT [.USER.FILES]

$ @SYS$LOGIN:DELTREE.COM [-]FILES

Confirm. Delete directory tree SYS$SYSDEVICE:[SMACK.USER.FILES] [N] Y

Confirm. Remove directory files too [Y] N

%DELETE-I-FILDEL, SYS$SYSDEVICE:[SMACK.USER.FILES]CLUE$STARTUP_.LOG;36 deleted (9 blocks)
%DELETE-I-FILDEL, SYS$SYSDEVICE:[SMACK.USER.FILES]CLUE$STARTUP_SWPCTX.LOG;91 deleted (9 blocks)
%DELETE-I-FILDEL, SYS$SYSDEVICE:[SMACK.USER.FILES]DECW$SERVER_0_ERROR.LOG;112 deleted (3 blocks)
%DELETE-I-FILDEL, SYS$SYSDEVICE:[SMACK.USER.FILES]ERT_ACTIVITY.LOG;66 deleted (0 blocks)
%DELETE-I-FILDEL, SYS$SYSDEVICE:[SMACK.USER.FILES]ESS$LAD_STARTUP.LOG;1 deleted (3 blocks)
%DELETE-I-FILDEL, SYS$SYSDEVICE:[SMACK.USER.FILES]ESS$LAST_STARTUP.LOG;1 deleted (27 blocks)
%DELETE-I-FILDEL, SYS$SYSDEVICE:[SMACK.USER.FILES]ESS$STARTUP.LOG;2 deleted (6 blocks)
%DELETE-I-FILDEL, SYS$SYSDEVICE:[SMACK.USER.FILES]NETSERVER.LOG;43 deleted (3 blocks)
%DELETE-I-FILDEL, SYS$SYSDEVICE:[SMACK.USER.FILES]SETHOST.LOG;3 deleted (2088 blocks)
```

A copy of DELTREE.COM is available on the OpenVMS AXO distribution media in the [DOCUMENTATION.V061.EXAMPLES] directory. For more information about how to access the files on the distribution media, see the *OpenVMS AXP CD–ROM User's Guide*.

A

New OpenVMS System Messages

This release includes new messages for the following OpenVMS facilities:

- ANALDISK, Analyze/Disk_Structure Utility
- BACKUP, Backup Utility
- DDTM, DECdtm Services
- ERF, Error Log Utility
- INSTALL, Install Utility
- LAT, LAT Facility
- LINK, Linker Utility
- LMCP, Log Manager Control Program
- LICENSE, License Management Utility
- MOUNT, Mount Utility
- MSGHLP, Help Message utility
- PCSI, POLYCENTER Software Installation Utility
- PCSIUI, POLYCENTER Software Installation Utility
- PRINT, PRINT Command
- PSM, Print Symbiont Modification Facility
- RMS, OpenVMS Record Management Services
- SDA, System Dump Analyzer
- SET, SET Command and SET Utility
- SHOW, SHOW Command
- SUBMIT, SUBMIT Command
- SYSTEM, System Services
- TIE, Translated Image Environment

This appendix alphabetically lists and describes all messages that have been added for this release. You can access online descriptions of these and all other OpenVMS system messages by using the online Help Message utility. For information about the HELP/MESSAGE command and qualifiers, see DCL help or refer to the *OpenVMS System Messages: Companion Guide for Help Message Users*. New Help Message features for this release are described in Section 2.5.

ABORT, fatal error encountered - operation terminated

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: An unrecoverable error occurred.

User Action: Query the product database to determine the cause of the error. Correct the problem and retry the operation.

ALRDYINST, product already installed; use RECONFIGURE to modify

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This product is referred to by another product in the current installation. The product will not be reinstalled.

User Action: None. If you want to modify the product, you must use the RECONFIGURE command.

APPLYTOERR, product 'product-name' to which maintenance product 'product-name' applies not found

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You attempted to patch or apply a mandatory update to a product that cannot be found on the system.

User Action: Verify that you have the correct mandatory update or patch kit and that the product you want to update is on the system.

ASKABORT, Terminating is strongly recommended. Do you want to terminate? [YES]

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: A severe error was detected by the POLYCENTER Software Installation utility during the operation. Continuing might leave the system in an incomplete state, for example, with partially installed software. Terminating leaves the product database in the state it was in before this operation was attempted.

User Action: Terminate the operation and identify the problem before reattempting the operation. An accompanying error message provides more information about the problem.

BADREASON, abort reason code was invalid

Facility: SYSTEM, System Services

Explanation: The abort reason code passed in the **reason** argument of the $ABORT_TRANS service is invalid.

User Action: Use a valid abort reason code.

BADSYSRET, VAX return from 'address' to unsupported system address 'address'.

Facility: TIE, Translated Image Environment

Explanation: A translated RET instruction has an unsupported system address as its return address.

User Action: Fix the original OpenVMS VAX code to use only those system services supported by OpenVMS VAX systems.

Block Size - Total: 'number' Optional: 'number' Required: 'number'

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This message reports the total disk block consumption of a product or a product option.

User Action: None, unless total disk block consumption of the product exceeds the space available on the destination device. In this case, stop the operation and create enough disk space on the destination device.

CANCEL_WIP, termination resulted in an incomplete modification to the system

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You interrupted an operation that was executing, leaving the system in an incomplete state. For example, if you interrupt an installation in progress, the utility does not remove any files created by the installation.

User Action: Digital recommends that you do not interrupt POLYCENTER Software Installation utility operations.

CANTRUNACP, attempt to run LATACP incorrectly

Facility: LAT, LAT Facility

Explanation: An attempt was made to run the LATACP process interactively. This action is prohibited, as it can cause LATACP to start improperly.

User Action: Use the LAT$STARTUP command procedure to automatically start both LAT and LATACP.

COMPWERR, operation completed after explicit continuation from errors

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: If you answer No to the ASKABORT query after a severe error occurs, this message displays when the operation completes.

User Action: Query the product database (PDB) to determine the cause of the error. Correct the problem and retry the operation.

CONFIG, product has already been configured

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This product has already been configured during this operation, either because it was selected or because it was included due to a dependency of another product.

User Action: None.

CONNDELETEONLY, connection can only be deleted

Facility: LAT, LAT Facility

Explanation: You tried to change a virtual circuit characteristic. This information cannot be changed.

User Action: Do not attempt to change virtual circuit characteristics.

CONNECTQUEUED, queued at service position 'position'

Facility: LAT, LAT Facility

Explanation: A LAT connection request issued to a service with no available resources has been placed in the remote node's service connection queue. When resources become available, the connection request will be completed.

User Action: Wait until the connection request is serviced. To abort the queued connection, press Ctrl/Y.

CONREMHV, confirm removal of higher version of 'product-name'

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: This message asks you to confirm whether to remove the higher version of the specified product.

User Action: Enter Yes to remove the higher version of the product. Enter No to keep the higher version and to omit the lower version on the system.

CONREMOS, confirm removal of operating system 'product-name'

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: This message displays with a question about whether to remove the specified operating system.

User Action: Confirm that you want to remove the operating system or else exit the operation.

CONREMUNR, optionally remove product 'product-name' that is no longer required by another product

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A software product that you are removing depends on another software product. This message asks if you want to remove the dependent software product.

User Action: Enter Yes to remove the specified product. Enter No or press Return to keep the product.

COPYREQD, destination specification required for this operation

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: No destination directory was specified.

User Action: Use the /DESTINATION qualifier on the COPY command to specify a destination directory for the copy of the product kit.

CURTIDCHANGE, default transaction currently being changed

Facility: SYSTEM, System Services

Explanation: An attempt was made to change the default transaction while a change is already in progress.

User Action: Retry the transaction.

DBINUSE, product database in use; waiting...

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: Another user is accessing the product database (PDB).

User Action: None. The utility processes your request when the current user is finished using the PDB.

DEFER, that item has been deferred. Please set the default value

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: No value is assigned to this configuration choice in the *input* configuration file.

User Action: Supply a default value. This default configuration choice will be presented to the user during installation.

DELENTRY, deleted queue entry 'id'

Facility: LAT, LAT Facility

Explanation: The specified queue entry ID was deleted.

User Action: None.

DELETEDCONN, deleted connection 'connect-id'

Facility: LAT, LAT Facility

Explanation: A delete circuit operation completed successfully.

User Action: None.

Do you want all the defaults for the subitems?

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This question prompts you to select the default values or view the configuration choices.

User Action: Enter Yes or No:

Yes	Automatically selects all of the defaults established by either the product or the product database. You will not see any configuration choices nested inside this configuration choice.
No	Displays the configuration choices that you can select.

DUPPROD, duplicate product ignored

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: The product has already been selected for the operation.

User Action: None.

ENTRYDELONLY, queue entry can only be deleted

Facility: LAT, LAT Facility

Explanation: An attempt was made to use a SETMODE $QIO operation on a queue entry. You can only delete queue entries from the system or use SENSEMODE $QIO operations on them.

User Action: Do not use SETMODE $QIO operations on queue entries.

ERRNOAPPLY, APPLY TO statement is required for this distribution type

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You are using a mandatory update or patch distribution kit that does not contain an *apply to* statement.

User Action: Add an *apply to* statement to the kit. A mandatory update or patch kit must name the product to which the patch or update applies.

ERRPDFPRD, product description file name does not match PRODUCT statement

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The file name of the product description file (PDF) does not match the information contained in the *product* statement.

User Action: Make sure the file name of the PDF and the *product* statement contain the same information.

EXEFAIL, execute statement failed; status returned from DCL follows

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: An *execute* statement failed.

User Action: Take action based on the accompanying message.

FAILCONF, failed to resolve conflicting requirements for 'object-type' 'object-name'

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: More than one product supplies a managed object of the same name, type, and scope. They cannot coexist because of a conflict in the objects' properties.

User Action: Determine which product's managed object you want to preserve and make the necessary adjustments to the system.

FAILCONFNNM, failed to resolve conflicting requirements for 'object-type' (unnamed)

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: More than one product supplies a managed object of the same name, type, and scope. They cannot coexist because of a conflict in the objects' properties.

User Action: Determine which product's managed object you want to preserve and make the necessary adjustments to the system.

FLTINE_F, floating inexact fault, PC='xxxxxxxx', PS='xxxxxxxx'

Facility: SYSTEM, System Services

Explanation: A floating arithmetic or conversion operation gave a result that differs from the mathematically correct result.

User Action: Examine the PC location displayed in the message. Check the program listing to verify that the operands or variables are specified correctly.

FLTINV_F, floating invalid fault, PC='xxxxxxxx', PS='xxxxxxxx'

Facility: SYSTEM, System Services

Explanation: One or more illegal operand values were detected on an attempt to perform a floating-point arithmetic, conversion, or comparison operation.

User Action: Examine the specified PC location. Check the program listing to verify that all operands or variables are specified correctly.

FORCED, product 'product-name' implicitly selected

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The named product has been implicitly selected to execute in the current operation.

User Action: None.

FORCEDDES, product 'product-name' implicitly deselected

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The named product has been implicitly deselected so that it will not execute in the current operation.

User Action: None.

FORCEDRCF, product 'product-name' implicitly selected for reconfiguration

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The named product has been implicitly selected to be reconfigured in the current operation.

User Action: None.

FORCEDUPG, product 'product-name' implicitly removed

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The named product has been implicitly selected to be removed in the current operation.

User Action: None.

Global Pages - Total: 'number' Optional: 'number' Required: 'number'

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This message reports the global page consumption of a product or a product option.

User Action: None, unless global page consumption of the product exceeds a limit set by a system or process parameter. In this case, modify the system or process parameters accordingly.

Global Sections - Total: 'number' Optional: 'number' Required: 'number'

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This message reports the global section consumption of a product or product option.

User Action: None, unless global section consumption of the product exceeds a limit set by a system or process parameter. In this case, modify the system or process parameters accordingly.

IDXSEARCH, key search encountered index structure inconsistency - convert file

Facility: RMS, OpenVMS Record Management Services

Explanation: The last keyed access to the file exceeded a maximum iteration count for restarting the operation at the root index bucket for the key of reference. This condition can occur if the index structure has become inconsistent. The status value (STV) of the RAB contains the key of reference.

User Action: Convert the file and retry the operation. If the error persists, contact a Digital service representative.

ILLEGAL_SHADOW, illegal formed trap shadow, Imask='xxxxxxxx', Fmask='xxxxxxxx', summary='xx', PC='xxxxxxxx', PS='xxxxxxxx'

Facility: SYSTEM, System Services

Explanation: The instruction sequence does not obey the rules stated in the *Alpha Architecture Reference Manual* for software completion. It is impossible for system software to complete the IEEE floating-point arithmetic operation.

User Action: Recompile the program to enable software completion.

IMGREQ, qualifier 'qualifier' requires that an image be produced qualifier ignored

Facility: LINK, Linker Utility

Explanation: The specified qualifier cannot be used in a LINK command because it requires that the linker operation produce an image file.

User Action: Reissue the LINK command with the appropriate /EXECUTABLE, /SHAREABLE, or /SYSTEM qualifier.

INCOMPDRVACP, incompatible LATACP and LTDRIVER - LATACP terminating

Facility: LAT, LAT Facility

Explanation: An attempt to start LAT failed because of a mismatch between the LAT driver and LATACP.

User Action: Locate the mismatched image and remove it from the system.

INTEGRITY, transaction aborted because a resource manager integrity constraint check failed

Facility: DDTM, DECdtm Services

Explanation: Either you entered invalid data, or the resource manager integrity constraint check was not properly defined.

User Action: Verify your data, then retry the transaction.

INTOVF_F, integer overflow fault, PC='xxxxxxxx', PS='xxxxxxxx'

Facility: SYSTEM, System Services

Explanation: An integer overflow occurred on an OpenVMS AXP CVTQL or CVTGQ instruction, and integer overflow reporting is explicitly enabled in the instruction.

User Action: Examine the PC location displayed in the message. Check the program listing to verify that the operands or variables are specified correctly.

INVALKEY, that key is not valid

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: The answer or key you entered is not a valid reply to the question being asked.

User Action: Try another response.

INVALRESP, invalid response

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: Your response to the prompt is invalid.

User Action: Specify the number that is displayed to the left of the product you want to select.

INVAPLTO, multiple APPLY TO statements

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The product description file (PDF) contains more than one *apply to* statement.

User Action: Use only one *apply to* statement in the PDF.

INVPDFTYPE, APPLY TO statement requires distribution type of PATCH or MANDATORY UPDATE

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: An *apply to* statement occurs in a product description file (PDF) that is not a patch or mandatory update.

User Action: Either remove the *apply to* statement or change the product distribution kit type in the *product* statement to *patch* or *mandatory update*.

INVREMDST, remote destination must be mounted privately by the process performing the operation

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The POLYCENTER Software Installation utility tried to place material on a volume whose owner is not the same as the owner of the process.

User Action: Have the process owner privately mount the destination volume.

INVUPGRADE, multiple UPGRADE statements

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The product description file (PDF) contains more than one *upgrade* statement.

User Action: Use only one *upgrade* statement in the PDF.

'number' items found

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This message reports the number of matches found in the product database in response to a PRODUCT SHOW command.

User Action: None.

LASTIPCF, multiple input configuration files were specified: values in a later one may supersede those in a previous one

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This message displays whenever you specify a list of files with the /CONFIGURATION=INPUT qualifier to the PRODUCT command. The POLYCENTER Software Installation utility does not check whether a product configuration choice appears in any or several of the specified configuration files.

User Action: None required. You can inspect the configuration files to determine which products' values are contained in each.

LATSYM, LAT protocol requested on device 'device-name' which is not a LAT device

Facility: PSM, Print Symbiont Modification Facility

Explanation: This device is not a LAT device, but LAT symbiont protocol is being requested by the requested processor symbiont.

User Action: Either use a non-LAT symbiont for this device or choose a LAT device for this processor.

LOG_BUSY, transaction log busy

Facility: LMCP, Log Manager Control Program

Explanation: The dump aborted because the transaction log was too busy. This can happen when the transaction rate is very high.

User Action: Wait until the system is less busy, then try again.

LOG_FAIL, transaction aborted because an attempt to write to the transaction log failed

Facility: DDTM, DECdtm Services

Explanation: The DECdtm transaction log could not be written to.

User Action: Check whether the transaction log's disk is mounted. Use the Analyze/Disk_Structure utility to check for disk errors.

LOG_NOT_IN_USE, transaction log not in use

Facility: LMCP, Log Manager Control Program

Explanation: The transaction log is not in use and the TP_SERVER process is not running.

User Action: None.

LRJREMDISABLED, remote node is disabled

Facility: LAT, LAT Facility

Explanation: A LAT $QIO connection request failed because the target node has disabled LAT connections.

User Action: Enable connections on the remote node and retry the connection.

MISSLNKQUAL, image was not linked with /SECTION_BINDING=CODE

Facility: INSTALL, Install Utility

Explanation: The image was not linked using the qualifier /SECTION_BINDING=CODE. Specifying this qualifier ensures that the linker does not optimize calls between code image sections by using a relative branch instruction.

User Action: Either relink the image using the qualifier /SECTION_BINDING=CODE or do not install the image as resident.

MULOBJ, the 'object-type' 'object-name' is multiply defined; conflict resolution may occur

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: More than one managed object of the same name and type was detected.

User Action: None. The POLYCENTER Software Installation utility attempts to resolve the conflict.

MULOBJNNM, the 'object-type' (unnamed) is multiply defined; conflict resolution may occur

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: More than one managed object of the same name and type was detected.

User Action: None. The POLYCENTER Software Installation utility attempts to resolve the conflict.

NETBTS, network buffer too small for 'size' byte record

Facility: RMS, OpenVMS Record Management Services

Explanation: The network buffer is too small for an attempted $GET, $PUT, or $UPDATE operation on a remote file. For a $GET, no record is returned to the user's record buffer; however, the next record pointer is properly advanced to the next record in the remote file, if there is one, or to the end-of-file. The status value (STV) in the RAB contains the size of the record that is too large for the network buffer.

User Action: Increase the size of the network buffer to accommodate the maximum record size in the remote file. The default system setting for the SYSGEN RMS_DFNBC (network block count) parameter is 8, which becomes the process default if it is not overridden by the process.

Determine the size of the largest record in the file either by using the DCL command DIRECTORY/FULL or ANALYZE/RMS_FILE or, from within the program, by examining the contents of the XABFHC field XAB$W_LRL after opening the file with the XABFHC chained to the FAB.

Increase the network buffer size for the process by executing the DCL command SET RMS_DEFAULT/NETWORK_BLOCK_COUNT=n on the local node or, from within the program, by setting the XABITM item XAB$_NET_BLOCK_COUNT to 'n' when opening the file with the XABITM chained to the FAB. Determine the value of 'n' as follows:

- For a remote file opened from an image:

  n = maximum record size in blocks

  Example: For a maximum record size of 4112 bytes, n = 9.

  4112 divided by 512 = 8.03

- For a remote file opened using the DCL command OPEN, the actual size of the network buffer is derived by dividing the NETWORK_BLOCK_COUNT by 8. To compensate for this, the block size must be multiplied by 8:

  n = (maximum record size in blocks) x 8

  Example: For a maximum record size of 2048 bytes, n = 32.

  2048 divided by 512 = 4 x 8 = 32

NOACTLINKS, no active LAT links to service request

Facility: LAT, LAT Facility

Explanation: A LAT connection to the local node was attempted from the local node. The operation failed because LAT cannot communicate with the datalink layer.

User Action: Ask your system manager to properly create LAT$LINK and make it available.

NODATABIND, image was not linked with /SECTION_BINDING=DATA

Facility: INSTALL, Install Utility

Explanation: The image was not linked using the qualifier /SECTION_BINDING=DATA. Specifying this qualifier installs the image as data resident to ensure that no relative offsets exist between data image sections.

User Action: Either relink the image using the qualifier /SECTION_BINDING=CODE or do not install the image as resident.

NODEFER, defer key is not available at this time

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: No value was specified for the configuration choice default.

User Action: Enter a configuration choice value to complete the operation.

NODELSHRADR, unable to delete image with shareable address data

Facility: INSTALL, Install Utility

Explanation: The Install utility cannot delete an image that was installed with shareable address data. The following images are automatically installed this way when the SYSGEN parameter IMGREG_PAGES is nonzero: DECC$SHR, DPML$SHR, LIBRTL, LIBOTS, and CMA$TIS_SHR.

User Action: You must reboot the system to replace an image installed with shareable address data. If in the future you want to replace the images listed above without rebooting the system, you can set the IMGREG_PAGES parameter to 0 before you reboot.

NOFILENAME, an image in the EXEC list does not have a valid file name; not read

Facility: SDA, System Dump Analyzer

Explanation: SDA found an image with no file name while processing the READ/EXECUTIVE command. This condition occurs when the directory and file specification exceed the system loader's 39-character limit.

User Action: Reload the system image using a shorter directory and file specification.

NOGHREG, insufficient memory in the code or data granularity hint region

Facility: INSTALL, Install Utility

Explanation: An image was installed using the /RESIDENT qualifier, but the code or data granularity hint region does not have enough memory for the image's code or read-only data sections.

User Action: Consider adjusting the granularity hint region SYSGEN parameters: GH_EXEC_CODE, GH_EXEC_DATA, GH_RES_CODE, GH_RES_DATA, and GH_RSRVPGCNT.

NOHELP, there is no help available at this time

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: The Help key is not supported on this input device. Help is not available for individual items.

User Action: To access help information for this product, reenter the PRODUCT command with the /HELP qualifier.

NOITEMS, there are no items to answer for this product

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: The product has no configuration choices.

User Action: None.

NOLIBRARIES, none of the specified libraries were found; check logical name MSGHLP$LIBRARY

Facility: MSGHLP, Help Message utility

Explanation: The search path logical name MSGHLP$LIBRARY is defined to point to a number of database files, none of which exists.

User Action: Verify that the logical name search path defines existing Help Message database files, or deassign the logical name so that Help Message accesses the default database file, SYS$HELP:MSGHLP$LIBRARY.MSGHLP$DATA.

NOMATCH, no products found matching:

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: No products were found to satisfy the selected product filter or search criteria for the chosen operation.

User Action: None required. However, you can change the filters as desired to control product selection.

The /SOURCE qualifier (which defaults to the logical name PCSI$SOURCE) determines the source search path used to find products for operations that require kits. The POLYCENTER Software Installation utility searches the product database for other operations. The product name parameter and the /PRODUCER, /BASE_SYSTEM, and /VERSION qualifiers operate as product filters.

NOMOREENTRIES, no more entries in queue

Facility: LAT, LAT Facility

Explanation: There are no more entries in the LAT connection queue.

User Action: None.

NONRES, installed image non-resident with other specified options

Facility: INSTALL, Install Utility

Explanation: The image has been successfully installed, but not as a resident image.

User Action: None.

NOOPENIGN, /NOOPEN ignored: /SHARED or /HEADER_RESIDENT are enabled

Facility: INSTALL, Install Utility

Explanation: The /NOOPEN qualifier is specified with /SHARED or /HEADER_RESIDENT in a CREATE, ADD, or REPLACE command. The /NOOPEN qualifier is ignored.

User Action: None.

NOPRIV, insufficient privilege to display complete information

Facility: SHOW, SHOW Command

Explanation: You do not have sufficient privilege to see all the users of an Activity license.

User Action: See your system manager. You must obtain GROUP privilege to see license users in the same group and WORLD privilege to see license users outside your group.

NOPROD, no products found on which to perform this operation

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: No products were found to satisfy the selected product filter or search criteria for the chosen operation.

User Action: None required. However, you can change the filters as desired to control product selection.

The /SOURCE qualifier (which defaults to the logical name PCSI$SOURCE) determines the source search path used to find products for operations that require kits. The POLYCENTER Software Installation utility searches the product database for other operations. The product name parameter and the /PRODUCER, /BASE_SYSTEM, and /VERSION qualifiers operate as product filters.

NORESSEC, no resident sections created for 'file-name'

Facility: INSTALL, Install Utility

Explanation: The installed shareable image has no image sections that can be converted into resident sections.

User Action: Examine the image's linker map for changes that will separate shareable and nonshareable content into separate image sections.

NOSUCHNODE, node 'node-name' not known

Facility: LAT, LAT Facility

Explanation: A LAT connection was attempted to an unknown target node.

User Action: Retry the connection until the specified node is known, or check whether some condition on the specified node prohibits its network visibility. Possibly the local node's service group codes do not intersect with the group codes of the specified remote node.

NOTORIGIN, process did not start the transaction

Facility: SYSTEM, System Services

Explanation: The process calling $END_TRANS is not the same process that called $START_TRANS.

User Action: Check your application to ensure that $START_TRANS and $END_TRANS are called by the same process, then retry the transaction.

NOTTRANS, INFER statement requires distribution type of TRANSITION

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You are using an *infer* statement in a product description file (PDF) that is not a transition kit.

User Action: Either remove the *infer* statement or change the *product* statement to the *transition* type.

NOUNSHELVE, file 'file-name' is shelved, this process does not automatically unshelve files

Facility: PRINT, PRINT Command

Facility: SUBMIT, SUBMIT Command

Explanation: Your process tried to access a shelved file, and automatic unshelving is turned off.

User Action: Do one of the following:

- Use the DCL command SET PROCESS/AUTO_UNSHELVE to turn automatic unshelving on.
- Use the DCL command UNSHELVE to unshelve the shelved file.

NOWRITE, nowrite key is not available at this time

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: No configuration choice value was specified.

User Action: Specify a value for the configuration choice in order to complete the operation.

OBJDES, the 'object-type' 'object-name' is part of an option that was not selected - replacement skipped

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The named managed object will not execute in this operation.

User Action: None.

OBSPASSALL, terminal set to PASSALL mode even though /PASSALL qualifier is obsolete

Facility: SET, SET Command and SET Utility

Explanation: You specified the obsolete /PASSALL qualifier with the SET TERMINAL command.

User Action: Use the /PASSTHRU qualifier to perform this function.

OPREQRD, specify the operation (such as INSTALL) in the PRODUCT command line

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: A required parameter is missing from the command.

User Action: Specify the operation (for example, INSTALL, REMOVE, RECONFIGURE, SHOW) in the command line.

OPTINVKIT, OPTION statement invalid in distribution type PATCH, PARTIAL, MANDATORY UPDATE, or TRANSITION

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: An *option* statement is included in a product description file (PDF) that is not for one of the following kit types: full, operating system, or platform. The kit types named in the message do not allow you to present configuration choices to the user.

User Action: Either remove the *option* statement or modify the *product* statement to be the correct type.

OUTRANGE, number out of range

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: The response is not a number within the displayed range.

User Action: Enter an integer from the range of listed products.

PARDES, the 'object-type' 'object-name' is part of an option that was not selected - 'object-type' update skipped

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The named managed object will not execute in this operation.

User Action: None.

PART_SERIAL, transaction aborted because a resource manager serialization check failed

Facility: DDTM, DECdtm Services

Explanation: A resource manager serialization check failed.

User Action: Retry the transaction.

PART_TIMEOUT, transaction aborted because a resource manager timeout expired

Facility: DDTM, DECdtm Services

Explanation: A resource manager exceeded its execution timeout.

User Action: Retry the transaction.

PARUDF, the 'object-type' 'object-name' has not been provided by a previous Install or Register operation - 'object-type' update skipped

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You attempted to update a managed object that is not known to the POLYCENTER Software Installation utility, for example, an object provided by a previous install or register operation.

User Action: Check the product database and your command to verify that you are trying to update a known managed object.

PDFEND, unmatched end

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You specified a group statement in the product description file (PDF) without a corresponding *end* group statement.

User Action: Check the group statements in the PDF for missing or extraneous *end* statements.

PDFEOF, unexpected end of file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You specified a group statement in the product description file (PDF) without a corresponding *end* group statement.

User Action: Check the group statements in the PDF and add the missing *end* statement.

PDFLIN, 'line-number' 'text'

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: An error was detected in the specified line and text of the product description file (PDF).

User Action: Fix the text on the specified line in the PDF.

PDFMTX, duplicate or mutually exclusive option specified

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You either specified an option twice or specified incompatible options with a product description language (PDL) statement.

User Action: Review your option specifications.

PDFSTR, PRODUCT statement must be the only statement at the outermost level

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: There are statements outside the *product* and *end product* statements in the product description file (PDF).

User Action: Make sure all statements in the PDF come after the *product* statement and before the *end product* statement.

PKGCNT, packaging more than one product may result in an unexpected kit

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This informational message warns that kit problems can occur if more than one product's kit is packaged in one PACKAGE operation.

User Action: Use a separate PACKAGE command to package each kit.

PKGREQD, source, destination, and material specification required for this operation

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: The PACKAGE command is missing one or more required qualifiers.

User Action: Specify all of the following qualifiers in the PACKAGE command:

/SOURCE	Specifies an explicit source search path in which to find product description and text files.
/MATERIAL	Specifies a source search path for the files associated with the product.
/DESTINATION	Specifies an explicit destination directory specification in which to create the packaged kit.

PLSREBOOT, please reboot to install a new version of this image

Facility: INSTALL, Install Utility

Explanation: This informational message accompanies NODELSHRADR.

User Action: You must reboot the system to replace an image installed with shareable address data. If in the future you want to replace this image without rebooting the system, you can set the IMGREG_PAGES parameter to 0 before you reboot. Setting this parameter affects all of these images: DECC$SHR, DPML$SHR, LIBRTL, LIBOTS, and CMA$TIS_SHR.

PRCERR, error communicating with subprocess; status returned from DCL follows

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: An error occurred while communicating with the Digital Command Language (DCL) shell subprocess.

User Action: Take action based on any accompanying error messages and retry the operation.

PRDALRREM, product has been removed

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You attempted to remove a product that has already been removed.

User Action: Verify the name of the product you want to remove.

PRDSTAABS, operation invalid on product 'product-name' that is absent

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You attempted to perform an operation on a product that is not on the system or is not accessible to the POLYCENTER Software Installation utility.

User Action: Verify that you are attempting the operation on the correct product. If necessary, install the correct product and make it available on the system.

PTFDUPMOD, duplicate module name in product text file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: More than one text module has the same name in the product text file (PTF).

User Action: Remove the duplicate module or change the name of the module in the product text file (PTF).

PTFDUPPRM, duplicate parameter line for this module in product text file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: There is more than one parameter line (starting with an equal sign) of the same type for a text module in the product text file (PTF).

User Action: Change or remove any duplicate parameter lines in the text module.

PTFINVMOD, invalid module header syntax in product text file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A text module header in the product text file (PTF) contains a syntax error.

User Action: Fix the syntax of the text module header and the module name in the PTF.

PTFINVPRM, invalid parameter line syntax in product text file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: An unrecognized parameter line was found in a text module in the product text file (PTF).

User Action: Correct the syntax or remove the unrecognized parameter line.

PTFMISMAT, =PRODUCT directive field does not match PRODUCT statement

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The product information in the =PRODUCT directive of the product text file (PTF) does not match the product information in the *product* statement of the product description file (PDF).

User Action: Make sure the *product* statement and the =PRODUCT directive contain the same information.

PTFMISMOD, missing product text file module for 'name'

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The product description file (PDF) specifies a text module that does not exist in the product text file (PTF).

User Action: Verify that the PTF contains the correct text modules and that their names match those specified in the PDF.

PTFMISPRD, required =PRODUCT directive missing in product text file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The utility cannot find the =PRODUCT directive in the product text file (PTF).

User Action: Verify that the first line of the PTF is an =PRODUCT directive.

PTFMISPRM, missing required parameter line for previous module in product text file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A text module in the product text file (PTF) is missing a required parameter line (starting with an equal sign).

User Action: Review the matching statement types in the product description file (PDF) to determine what parameter lines are required in the text module. Add any missing parameter lines.

PTFNOHDR, missing module header in product text file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A text module in the product text file (PTF) is missing a module header.

User Action: Add a text module header to the module in the PTF.

PTFPRD1ST, =PRODUCT directive must be first line in product text file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The first line of the product text file (PTF) does not contain an =PRODUCT directive.

User Action: Correct the first line of the PTF.

PTFUNDMOD, undefined module name in product text file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A text module header in the product text file (PTF) has a module name that is not defined in the corresponding product description file (PDF).

User Action: Either remove the text module from the PTF or add the statement that corresponds to this text module to the PDF.

PTFUNDPRM, parameter line type not defined for this module in product text file

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A text module in the product text file (PTF) contains a parameter line (starting with an equal sign) that is not allowed for this module type.

User Action: Correct or remove the undefined parameter line.

REFLECT, Internal TIE exception code reported to user. Please submit an SPR.

Facility: TIE, Translated Image Environment

Explanation: The exception code reported is TIE$_REFLECT, an intermediate signal that should not be reported to the user.

User Action: Submit a Software Performance Report (SPR) that describes the circumstances under which the exception code was reported.

REFUNA, product 'product-name' is required by 'product-name'

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A product required by another product executing in this operation is unavailable.

User Action: Install the required product or make it available.

REFUNAPRC, product 'product-name' is required by 'product-name' on node 'node-name'

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A product required by another product executing in this operation is unavailable on a specific processor.

User Action: Install the required product or make it available on the processor.

RELOFIXDZ, deferred relocation/fixup is within a demand zero section at location 'location'

Facility: LINK, Linker Utility

Explanation: An attempt was made to write relocation or fixup information into an image section that is marked "demand zero." Demand zero sections must not have information written to them during the link process. This condition indicates a compiler or linker fault.

User Action: Submit a Software Performance Report (SPR).

RELOFIXLKG, relocation/fixup to a linkage section that is marked overlaid in psect 'psect-name' in module 'module-name' in file 'file-name'

Facility: LINK, Linker Utility

Explanation: An attempt was made to write relocation or fixup information into an image section that contains linkage information and is marked overlaid. Linkage sections cannot be overlaid.

User Action: Check whether you specified a PSECT_ATTRIBUTE options clause for a linkage section and marked it OVR. If so, delete the OVR attribute.

RELOFIXOUT, deferred relocation/fixup is outside image boundaries at location 'location'

Facility: LINK, Linker Utility

Explanation: An attempt was made to write relocation or fixup information into the image outside of the image section boundaries. This condition indicates a compiler or linker fault.

User Action: Submit a Software Performance Report (SPR).

REMRUNOS, cannot remove the running operating system

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: You attempted to remove the operating system without first shutting it down.

User Action: Shut down the operating system and retry the operation.

Requires 'number' disk blocks.

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This product or configuration choice requires the specified number of disk blocks.

User Action: None, unless disk space might be overrun by this operation. In this case, stop the operation and create enough disk space on the destination device.

Requires from 'number' to 'number' disk blocks depending upon configuration.

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: This product or configuration choice requires the specified number of disk blocks. The lower number is required if no configuration choices are selected. The higher number is required if all configuration choices are selected.

User Action: None, unless disk space might be overrun by this operation. In this case, stop the operation and create enough disk space on the destination device.

RESFAIL, failed to install image with /RESIDENT qualifier

Facility: INSTALL, Install Utility

Explanation: A create or replace operation failed.

User Action: Take action based on the accompanying message.

RESGEN, the 'object-type' 'object-name' provided by 'object-type' will remain installed; replacement material unavailable

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A managed object that has no replacement material in the newer version (or whose replacement material is unavailable) will remain on the system.

User Action: None.

RMTMNTFAIL, 'device' failed to mount on node 'node'

Facility: MOUNT, Mount Utility

Explanation: An attempt to mount a device clusterwide failed to mount the device on a remote node.

User Action: Take action based on the accompanying message.

RMVINVKIT, REMOVE statement invalid in distribution type TRANSITION

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A transition product description file (PDF) contains a *remove* statement.

User Action: Delete the *remove* statement or change the distribution type to something other than *transition*.

SERIALIZATION, transaction aborted because a serialization check failed

Facility: DDTM, DECdtm Services

Explanation: A serialization check failed.

User Action: Retry the transaction.

SFTCIRC, referenced product 'product-name' causes circular dependency

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: Two products refer to each other, causing a dependency that the utility cannot resolve.

User Action: Examine the products' product description files (PDFs) to ensure that they contain no circular *software* statement dependencies; for example, product A requires product B, which requires product A.

SFTCNFINS, installed product 'product-name' does not satisfy

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The specified installed product does not satisfy the constraints of a *software* statement in a selected product.

User Action: Terminate the operation and check the product database to be sure you are upgrading the correct product.

SFTCNFLCT, the following products also have requirements for 'product-name':

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: Certain selected or installed products have constraints on the named product.

User Action: None.

SFTCNFSEL, selected product 'product-name' does not satisfy

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The named, selected product does not satisfy the constraints of a *software* statement in another selected product.

User Action: Terminate the operation and deselect one of the conflicting products.

SFTREFERR, referenced product 'product-name' not found for product 'product-name'

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A product that is referenced by a selected product cannot be located.

User Action: Either deselect the product or locate the product kit (possibly in another source area) and try again.

SFTREFREM, referenced product 'product-name' removed from database

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The named, referenced product will be removed from the product database or, in some operations, the POLYCENTER Software Installation utility will ask whether to remove it.

User Action: If given a choice, choose Yes to remove the product or No to keep it.

SHELFERROR, access to shelved file failed

Facility: SYSTEM, System Services

Explanation: An attempt to access a file failed. The file was shelved and the shelving facility could not unshelve it.

User Action: Take action based on the accompanying message.

SHELVED, 'file-name' data not copied, file is shelved

Facility: BACKUP, Backup Utility

Explanation: An image mode save operation was performed on a file that is currently shelved. Therefore, only the file header was saved; no data was transferred to the save set.

User Action: None required. However, if you want to create an image backup that includes the contents of the specified file, use the UNSHELVE command to bring the file back on line before performing the image mode backup.

SHELVED, file is shelved

Facility: SYSTEM, System Services

Explanation: Your process tried to access a shelved file, and automatic unshelving is turned off.

User Action: Do one of the following:

- Use the DCL command SET PROCESS/AUTO_UNSHELVE to turn automatic unshelving on.
- Use the DCL command UNSHELVE to unshelve the shelved file.

SHLVSTATE, 'file-name' has changed shelved state

Facility: BACKUP, Backup Utility

Explanation: During a compare or verify operation, the Backup utility detected that the data contents of the specified file have been shelved and the file contents are unavailable for verification. This condition can occur when shelving activity on the volume causes the file state to change from online to shelved.

User Action: Specify the /AUTOUNSHELVE qualifier on the SET PROCESS command to change the process shelving default to autounshelve, or disable shelving activity during the backup operation.

SIDNOTVLD, SID qualifier valid only on VAX

Facility: ERF, Error Log Utility

Explanation: The SID qualifier cannot be used on the AXP version of the event report formatter.

User Action: Reenter the command using the MODEL= qualifier to specify the hexadecimal model number.

TARGET_GOTO_UNWIND, target invocation for $GOTO_UNWIND

Facility: SYSTEM, System Services

Explanation: The condition handler for the target invocation of a $GOTO_UNWIND operation has been notified that the target invocation frame is about to be unwound.

User Action: None.

TARGET_UNWIND, target invocation for $UNWIND

Facility: SYSTEM, System Services

Explanation: The condition handler for the target invocation of an $UNWIND operation has been notified that the target invocation frame is about to be unwound.

User Action: None.

TERMIMM, 'number' licenses will terminate in 'number' days

Facility: SHOW, SHOW Command

Explanation: Termination is imminent for one or more of the listed licenses.

User Action: If you plan to continue using these products, you must obtain a new license before the specified termination date to maintain product availability.

UNDEFENTRY, invalid queue entry request ID

Facility: LAT, LAT Facility

Explanation: The queue entry you specified does not exist.

User Action: Use the LATCP SHOW QUEUE_ENTRY command to list all entries in the local queue. Retry the command, using a valid queue entry.

UNDEFER, that item is no longer deferred. Please set the value

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: No value is set for this configuration choice.

User Action: Supply a value for the configuration choice if the choices are to be written to a configuration file.

UNRDEP, failed to resolve execution order dependency for 'object-type' 'object-name'

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A required software dependency cannot be resolved.

User Action: Check the product database and available kits for more information.

UNRDEPNNM, failed to resolve execution order dependency for 'object-type' (unnamed)

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A required software dependency cannot be resolved.

User Action: Check the product database and available kits for more information.

UNSHELVED, file is not shelved

Facility: SYSTEM, System Services

Explanation: You tried to unshelve a file that is not shelved.

User Action: None.

UNSHELVERR, error unshelving file 'file-name'

Facility: PRINT, PRINT Command

Facility: SUBMIT, SUBMIT Command

Explanation: An RMS error or a memory management error occurred while unshelving the file.

User Action: Take action based on the accompanying messages.

UNSUPCOMBOP, 'product-name' uses an unsupported combination of options and license type

Facility: LICENSE, License Management Utility

Explanation: You attempted to load a license with attributes that LMF does not support. Specifically, the combination of key options and license type is not supported.

User Action: Notify the issuer of the Product Authorization Key (PAK) and obtain a PAK that LMF can use.

UNVSTATUS, translation of symbol $STATUS is unavailable

Facility: MSGHLP, Help Message utility

Explanation: Help Message cannot translate the CLI symbol $STATUS to determine the result of the last executed command. Either the $STATUS symbol has been deleted, no command has been executed, or the first and only command was aborted by Ctrl/Y.

User Action: Reissue the HELP/MESSAGE command and specify a search string or the /STATUS or /FACILITY qualifier. The default HELP/MESSAGE command will work properly when $STATUS is available and a command has been executed.

UPGRADERR, existing version of product that 'product-name' upgrades is incorrect

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: The named product has an *upgrade* statement that directs the POLYCENTER Software Installation utility to upgrade an existing version of the same product. The existing version fails to meet the version constraints of the new product. An upgrade of the existing product is not possible with this product kit.

User Action: Verify whether the upgrade was already completed.

VERERR, verification error for 'file-name'

Facility: PCSI, POLYCENTER Software Installation Utility

Explanation: A verification error occurred on the named file. The resulting output file is not equal to the input file.

User Action: Retry the operation.

VERSELCT, this selection unselected a previous version

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: The current product version selection has superseded a previous product version selection. Only one version of a product can be selected for this operation.

User Action: Verify that the correct product version is selected.

VOLSETALRMNT, another volume set of same label already mounted

Facility: MOUNT, Mount Utility

Explanation: Either a volume set with the specified name is already mounted in the cluster, or the devices were mounted into the volume set in a different order on another node.

User Action: Rename the volume set to a unique name or, if the same volume set is already mounted on another node, ensure that devices are mounted into the volume set in the same order.

WRGARCH, 'product-name' license is not valid on this architecture

Facility: LICENSE, License Management Utility

Explanation: You attempted to load a license that cannot be used on the current architecture. For example, you attempted to load a license designated for use only on VAX systems onto an AXP system.

User Action: Obtain the correct Product Authorization Key (PAK) for your system.

WRITE, that item will be written to configuration file. Please set the default value

Facility: PCSIUI, POLYCENTER Software Installation Utility

Explanation: No value is assigned for this configuration choice.

User Action: Supply a value for the configuration choice if this choice is to be written to a configuration file. If no file is being written, you can press Ctrl/D to defer the choice.

Index

A

Access control
  protected objects, 3–11
  protected subsystems, 3–9, 3–14
  SET SECURITY command, 3–9
  SHOW SECURITY command, 3–12
Access control entries
  See ACEs
Access types
  LIB$ conversion routines, 3–13
  protected objects, 3–13
AccessX support, 5–6
ACEs (access control entries)
  Alarm ACE, 3–11
  Audit ACE, 3–11
  Creator ACE, 3–15
ACL editor, 3–16
/ADDRESS qualifier
  with the SET PROCESS command, 4–15
  with the SHOW PROCESS command, 4–15
AGEN$INCLUDE_PARAMS files, 3–2
AGEN$NEW_NODE_DEFAULTS.DAT file, 3–2
AGEN$NEW_SATELLITE_DEFAULTS.DAT file, 3–2
Alarm ACE, 3–11
Alarm events, 3–10
Associate access, 3–13
Atomic data types
  X_floating, 4–16
Attention AST, 4–30
Attributes for identifiers
  Holder Hidden, 3–15
  Name Hidden, 3–15
  No Access, 3–15
  Subsystem, 3–15
/ATTRIBUTES qualifier
  in Linker utility, 4–13
Audit ACE, 3–11
Audit Analysis utility (ANALYZE/AUDIT), 3–16
Audit events, 3–10
Auditing
  alarm events, 3–10
  audit events, 3–10
  controlling messages, 3–9, 3–11
  management interface, 3–10
  new events, 3–10
Auditing (cont'd)
  object access, 3–11
  object ACEs, 3–11
  support for objects, 3–10
AUDIT privilege, 3–16
Audit server startup, 3–16
AUTOGEN parameter files, 3–2

B

Backing up the system disk, 3–5
BACKUP command
  backing up the system disk, 3–5
  /EXACT_ORDER qualifier, 3–6
  restoring the system disk, 3–5
Batch and print queuing system features, 3–8
  customizing LAT symbionts, 4–29
  $GETQUI system service, 4–22
  multiple queue managers, 3–8
  $SNDJBC system service, 4–22
Booting
  cross-architecture, 3–2
Bucket reclamation statistics, 4–11

C

C
  data type declarations, 4–18
  implementation table, 4–18
C++
  data type declarations, 4–18
  data types, 4–18
  implementation table, 4–18
Cache
  virtual I/O, 3–4
$CHKPRO system service, 3–10
CLUE (Crash Log Utility Extractor)
  See Crash Log Utility Extractor
/CLUSTER qualifier
  with the SHOW LICENSE command, 3–4
Common event flag cluster protection, 3–11
Complex_number data type, 4–17
Configurations
  cross-architecture booting, 3–2
Context argument
  to $GETQUI system service, 4–22

Control access, 3–13
CONV$RECLAIM key selection parameter, 4–9
CONVERT command
  list of qualifiers, 4–9
CONVERT conversion statistics, 4–10
CONVERT/RECLAIM /KEY qualifier, 2–10
Crash Log Utility Extractor (CLUE)
  commands
    archiving information, 4–14
Create access, 3–13
$CREATE_USER_PROFILE system service, 3–13
Creator ACE, 3–15
Customizing the display server, 5–1

D

Data types
  atomic
    DSC$K_DTYPE_FX, 4–16
    DSC$K_DTYPE_FXC, 4–16
  C++ declarations, 4–18
  C++ programming, 4–18
  C declarations, 4–18
Date/time formats, 2–10
DBGTK_SCRATCH system parameter, 3–17
DCL commands
  Common DCL commands, 2–1
  DEBUG command, 2–2 to 2–3
  DIAGNOSE command, 2–3
  DIFFERENCES command, 2–3
  DIRECTORY command, 2–3
  displaying updated information about processes, 2–7
  DUMP command, 2–3
  F$FILE_ATTRIBUTES lexical function, 2–4
  F$FILE_GETDVI lexical function, 2–4
  former platform-specific (now common) DCL commands, 2–1
  HELP command, 2–4
  Kept Debugger, 2–2
  output navigation qualifiers, 2–8
  PRODUCT command, 2–4
  RECALL command, 2–4
  REPLY command, 2–5
  SEARCH command, 2–5
  SET AUDIT command, 2–5
  SET FILE command, 2–5
  SET HOST/LAT command, 2–5
  SET PROCESS command, 2–6
  SET SECURITY command, 2–6
  SET TERMINAL command, 2–7
  shelving, 2–5, 2–6, 2–7
  SHOW AUDIT command, 2–7
  SHOW DEVICES command, 2–7
  SHOW ENTRY command, 2–7
  SHOW ERROR command, 2–7
  SHOW LICENSE command, 2–7
  SHOW LOGICAL command, 2–7
DCL commands (cont'd)
  SHOW MEMORY command, 2–7
  SHOW PROCESS command, 2–7
  SHOW QUEUE command, 2–8
  SHOW SECURITY command, 2–8
  SHOW SYSTEM command, 2–8
  SHOW USERS command, 2–8
  TYPE command, 2–8
DE422 controller, 4–36
DE425 controller, 4–36
Debugging multiprocessing programs
  enhanced support, 4–6
Debug symbol file
  See also /DSF qualifier
  creating, 4–13
DECamds, 3–4
DEC C
  See C
DEC C++
  See C++
DECdtm system services, 4–23
DECdtm transaction log management, 3–7
DECdts service, 4–36
DECevent Event Management utility
  description, 3–7
  report types, 3–7
DECwindows Motif display server
  AccessX support, 5–6
  customizing, 5–1
  disability support, 5–6
  font compiler qualifier, 5–9
  font name support, 5–10
  font server support, 5–9
  keymap name support, 5–7
  parameters for customizing, 5–1
  server extensions, 5–4
  shared memory extension, 5–5
DECwindows Motif for OpenVMS software, 5–1
DEFAA controller, 4–36
DEFEA controller, 4–36
DEFQA controller, 4–36
Delete access, 3–13
DETRA controller, 4–36
Device characteristics
  LAN drivers, 4–43
Device protection enhancements, 3–14
Devices
  excluding, 3–20
Device support
  debugging device drivers, 4–1
  Step 2 driver interface, 4–1
  user-written drivers, 4–1
  writing drivers in C, 4–1
  writing drivers in Macro-32, 4–1
Digital availability management software
  See DECamds

Digital Portable Mathematics Library
See DPML
Display server
See DECwindows Motif display server
Documentation comments, sending to Digital, iii
DOWNGRADE privilege, 3–16
DPML
new routines, 4–15
/DSF qualifier
in Linker utility, 4–13
Dump file information
saving automatically, 4–14
DW425 controller, 4–36

E

ERRFMT
See Error formatter
ERRFMT$SEND_TO logical name, 3–7
ERRFMT$_SEND_MAIL logical name, 3–7
Error formatter (ERRFMT), 3–7
Error Log utility
/MODEL qualifier, 3–7
Error messages
new messages, A–1
Ethernet controllers, 4–36
Events
security auditing, 3–10
Exception condition
SS$_PAGRDERR, 4–36
Excluding devices, 3–20
Exclusion list
displaying, 3–20
Execute access, 3–13

F

Facility identifiers, 3–15
FDDI controllers, 4–36
FDL$CREATE
optional argument *dlft_fdl_spc*, 4–11
Feedback on documentation, sending to Digital, iii
File information block (FIB) changes, 4–27
File shelving
DCL commands, 2–5, 2–6, 2–7
general information, 2–11
programming information, 4–12
File system
performance improvements, 3–6
Floating point numbers
X_floating complex, 4–17
X_floating standard, 4–16
Floating_point data type, 4–16
Font compiler, 5–9
Fonts
in DECwindows display server, 5–10
Lucida Bright, 5–10
Lucida Sans, 5–10
Fonts (cont'd)
Lucida Typewriter, 5–10
MIT miscellaneous, 5–10
Type 1, 5–10
VT330, 5–10
Font server support, 5–9
Function codes, 4–43
Function modifiers
IO$M_CTRL, 4–52

G

$GET_SECURITY system service, 3–13
GH_EXEC_CODE system parameter, 3–17
GH_EXEC_DATA system parameter, 3–17
GH_RES_CODE system parameter, 3–17
GH_RES_DATA system parameter, 3–17
Granularity hints, 3–17

H

Help Message utility (MSGHLP)
accessing $STATUS values for uninstalled messages, 2–12
message section files, 2–12
order searched, 2–13
new features, 2–11
/SECTION_FILE qualifier usage, 2–12
Holder Hidden identifier attribute, 3–15

I

IMGREG_PAGES system parameter, 3–17
Implementation tables
C and C++, 4–18
IMPORT privilege, 3–16
Include files
AGEN$INCLUDE_PARAMS files, 3–2
Installation procedures
using the POLYCENTER Software Installation utility, 3–5
Invoking SDA by default, 4–14
IO$M_LT_QUE_CHG_NOTIF, 4–30
IO$M_SENSE_MAC functional modifier
used with IO$_SENSEMODE, 4–52
IO$M_SHOW_MAP functional modifier
used with IO$_SENSEMODE, 4–53
IO$M_SHOW_ROUTE functional modifier
used with IO$_SENSEMODE, 4–53
IO$M_UPDATE_MAP functional modifier, 4–49
IO$_CREATE function code, 4–27
IO$_MODIFY function code, 4–28
IO$_M_SET_MAC functional modifier, 4–48
IO$_READLBLK function code, 4–43
IO$_READPBLK function code, 4–43
IO$_READVBLK function code, 4–43

IO$_SENSEMODE function code, 4–52
IO$_SETCHAR function code, 4–46
IO$_SETMODE
  IO$M_ROUTE functional modifier, 4–50
IO$_SETMODE function code, 4–46
IO$_WRITELBLK function code, 4–44
IO$_WRITEPBLK function code, 4–44
IO$_WRITEVBLK function code, 4–44
IO SET EXCLUDE command
  in SYSMAN, 3–20
IO SHOW EXCLUDE command
  in SYSMAN, 3–20

J

Journaling, 3–1

K

Kept Debugger, 2–2
Keymap name support, 5–7

L

LAN drivers
  address
    destination, 4–44, 4–45
    multicast, 4–44
    physical, 4–44
    source, 4–44
    Token Ring, 4–37
  buffer
    receive, 4–44
  characteristics
    devices, 4–37, 4–43
    drivers, 4–37
  device characteristics, 4–43, 4–52
  exclusive mode, 4–42
  frame formats, 4–38
  function modifiers, 4–52
  I/O functions, 4–43, 4–44, 4–46, 4–52
  IEEE 802 packet format
    read function, 4–44
    write function, 4–44
  modify characteristics, 4–46
  packet format
    set mode parameters, 4–48
  parameter ID, 4–46
    packet format, 4–48
  port
    start, 4–46
  protocol type, 4–44, 4–45
    sharing, 4–42
  read function, 4–43
  sense mode function, 4–52
  set characteristics function, 4–46
  set controller mode
    P2 buffer, 4–46
LAN drivers
  set controller mode (cont'd)
    parameter ID, 4–46
    protocol type sharing, 4–42
  set mode function, 4–46
  shared default mode, 4–42
  shared with destination mode, 4–42
  SYS$GETDVI routine, 4–43
  write function, 4–44
Language support, 2–10
LAT$C_ENT_QUEUE_ENTRY, 4–31, 4–33
LAT $QIO, 4–30
LAT Control Program (LATCP) utility, 3–21
  calculating service rating, 3–24
  new commands and qualifiers, 3–22
  queuing incoming requests, 3–23
  setting up limited services, 3–23
  specifying wildcards, 3–25
  translating logical names, 3–24
LATCP (Local Area Transport Control Program)
  See LAT Control Program utility
LAT function modifier, 4–30
LAT SENSEMODE, 4–31
  node counters, 4–31
  port counters, 4–31
  port entities, 4–32
  protocol errors, 4–31
  queue entries, 4–32
  service counters subblock, 4–32
LAT SETMODE, 4–33
  node entities, 4–33
  port type, 4–35
  service type, 4–35
LAT software
  calculating service rating, 3–24
  creating intrasystem LAT connections, 3–21
  creating limited services, 3–23
  LAT Control Program (LATCP) utility, 3–21
  new qualifiers to SET HOST /LAT command, 3–22
  propagating dialup bit, 3–21
  queuing incoming requests, 3–23
  SET HOST /LAT command, 3–21 to 3–22
  specifying wildcards in LAT Control Program (LATCP) utility, 3–25
  translating logical names, 3–24
  using SET HOST /LAT in command procedure, 3–22
LAT symbionts
  customizing, 4–29
LAT transport, 5–6
  creating the service, 5–6
LGI_CALLOUTS system parameter, 3–16
LIB$ routines
  LIB$FORMAT_SOGW_PROT, 3–13
  LIB$GET_ACCNAM, 3–13
  LIB$GET_ACCNAM_BY_CONTEXT, 3–14
  LIB$PARSE_ACCESS_CODE, 3–14

LIB$ routines (cont'd)
LIB$PARSE_SOGW_PROT, 3–14
Linker utility
enhancements, 4–13
Lock access, 3–13
Logical access, 3–13
LOGINOUT
callout routines, 4–13
Log Manager Control Program (LMCP) utility
CLOSE LOG command, 3–7
Lucida font support, 5–10

M

MACRO-32 Compiler for OpenVMS AXP, 4–21
Manage access, 3–13
Messages
new system messages, A–1
Monitoring Performance History (MPH) program, 3–1
MOUNT/SUBSYSTEM command, 3–14
Movefile subfunction, 4–27
MPH (Monitoring Performance History program)
See Monitoring Performance History program
Multiple queue managers, 3–8

N

Name Hidden identifier attribute, 3–15
Nearline storage
definition, 2–11
NET_CALLOUTS system parameter, 3–16
No Access identifier attribute, 3–15

O

Object classes
auditing support, 3–10
Object protection, 3–11
Offline storage
definition, 2–11
Online storage
definition, 2–11
OpenVMS AXP System-Code Debugger
See System-code debugger
OpenVMS Debugger
CONNECT command, 4–6
debugging the OpenVMS AXP operating system, 4–6
DECwindows Motif user interface, 4–3
DEPOSIT command, 4–3
DISCONNECT command, 4–6
EXAMINE command, 4–3
floating-point arithmetic, 4–3
language support, 4–3
MONITOR command, 4–3
new features, 4–2
online help, 4–7
OpenVMS Debugger (cont'd)
REBOOT command, 4–6
resident images, 4–4
SET TYPE command, 4–3
static watchpoints, 4–5
support for system-code debugger, 4–6
OpenVMS RMS
See RMS

P

/PAGE=SAVE qualifier navigation keys, 2–9
Parameters
See System parameters
Performance
virtual I/O cache, 3–4
PFN_COLOR_COUNT system parameter, 3–17
Physical access, 3–13
POLYCENTER HSM for OpenVMS, 2–11, 4–12
POLYCENTER Software Installation utility, 3–5
adding files, 3–5
benefits of using, 3–5
installing the operating system, 3–5
removing files, 3–5
upgrading the operating system, 3–5
Privileges, 3–16
Program access to conversion statistics array, 4–10
Program access to Convert options array, 4–9
Program access to Convert/Reclaim statistics array, 4–11
Protected objects, 3–9
access conversion routines, 3–13
access types, 3–13
auditing support, 3–10
common event flag clusters, 3–11
control access, 3–13
Creator ACEs, 3–15
database, 3–12
device protection check, 3–14
protected subsystems, 3–14
resource domains, 3–11
security class, 3–11
security profiles, 3–12
SET SECURITY command, 3–9
SHOW SECURITY command, 3–9
supported classes, 3–12
volumes, 3–11
Protected subsystems, 3–9, 3–14

Q

Queue Entry argument, 4–31, 4–33
Queue managers
multiple, 3–8

Queuing system features
See Batch and print queuing system features
QUI$_DISPLAY_MANAGER function code
for $GETQUI system service, 4–22

R

RAB$B_KSZ
used with performance caching optimization, 4–8
RAB$M_EQNXT option, 4–7
used with reverse-search (RAB$M_REV) option, 4–7
RAB$M_KGT option
used with reverse-search (RAB$M_REV) option, 4–7
RAB$M_NXT option, 4–7
RAB$M_REV reverse-search option, 4–7
Read access, 3–13
Record Management Services
See RMS
Reliability data, 3–1
Resource domain protection, 3–11
Restoring files on the system disk, 3–5
Reverse-search key option
design characteristics, 4–8
performance, 4–8
performance caching optimization, 4–8
used with duplicate keys, 4–8
use restriction, 4–8
Rights identifiers
attributes, 3–15
facility, 3–15
RIGHTSLIST.DAT protection, 3–16
RMS
reverse-search option, 4–7
RMS Journaling
on OpenVMS AXP systems, 3–1
Run-time library routines
LIB$FORMAT_SOGW_PROT, 3–13
LIB$GET_ACCNAM, 3–13
LIB$GET_ACCNAM_BY_CONTEXT, 3–14
LIB$PARSE_ACCESS_CODE, 3–14
LIB$PARSE_SOGW_PROT, 3–14
Run-time library support, 4–21

S

Satellites
booting across architectures, 3–2
SDA (System Dump Analyzer)
See System Dump Analyzer utility
SDA CLUE commands
collecting dump file information, 4–14
Security
management changes, 3–16
new protected objects, 3–11
Security (cont'd)
object management interface, 3–11
object profiles, 3–12
protected objects, 3–9
summary of features, 3–9
Security auditing
alarm events, 3–10
audit events, 3–10
delivering event messages, 3–10
management interface, 3–10
new events, 3–10
object access, 3–11
privilege auditing, 3–10
Security class, 3–11
SECURITY_POLICY system parameter, 3–16, 3–18
Sense mode function, 4–52
Sequential access, 4–8
Server
See Display server
SET AUDIT command, 3–10
BACKLOG qualifier, 3–11
SET HOST /LAT command, 3–21 to 3–22
/AUTOPROMPT qualifier, 3–22
creating intrasystem LAT connections, 3–21
new qualifiers, 3–22
/NOAUTOPROMPT qualifier, 3–22
/NOQUEUE qualifier, 3–22
propagating dialup bit, 3–21
/QUEUE qualifier, 3–22
using in command procedure, 3–22
$SETIME system service, 4–36
SET MESSAGE command
used to access uninstalled messages, 2–12
Set mode function
IO$M_SET_MAC qualifier, 4–48
SET RIGHTS_LIST command, 3–16
SET SECURITY command, 3–9, 3–12
SET VOLUME/SUBSYSTEM command, 3–14
$SET_RESOURCE_DOMAIN system service, 3–13
$SET_SECURITY system service, 3–13
SHADOWING system parameter, 3–18
SHADOW_MAX_COPY system parameter, 3–19
SHADOW_MBR_TMO system parameter, 3–19
SHADOW_SYS_DISK system parameter, 3–19
SHADOW_SYS_TMO system parameter, 3–17
SHADOW_SYS_UNIT system parameter, 3–19
SHADOW_SYS_WAIT system parameter, 3–17
Shared memory extension, 5–5
Shelving
DCL commands, 2–5, 2–6, 2–7
general information, 2–11
programming information, 4–12
SHOW AUDIT command, 3–10
SHOW LICENSE command, 3–4

SHOW SECURITY command, 3–9, 3–12
Shutdown
  saving cluster-visible security profiles, 3–12
SJC$_DELETE_QUEUE_MANAGER function code
  for $SNDJBC system service, 4–22
SS$_BADPARAM return, 4–42
SS$_DEVACTIVE, 4–30
SS$_DEVREQERR, 4–30
SS$_PAGRDERR, 4–36
$STATUS values
  accessing for uninstalled messages, 2–12
Submit access, 3–13
Subsystem ACE, 3–14
Subsystem attribute, 3–14
Subsystem identifier attribute, 3–15
SYS$AUDIT_EVENT system service, 3–10
SYS$CHECK_ACCESS system service, 3–11
SYS$CHECK_PRIVILEGE system service, 3–10
SYS$GETDVI routine
  LAN drivers, 4–43
SYS$GETJPI system service, 4–12
SYS$QIO system service, 4–12
SYS$SETSHLV system service, 4–12
SYS$SUBSYSTEM system service, 3–14
SYSGEN parameters
  See System parameters
SYSMAN commands, 3–20
System-code debugger
  use of, 4–1
System Dump Analyzer (SDA) utility
  CLUE commands, 4–14
  SET PROCESS command, 4–15
  SHOW PROCESS command, 4–15
System management examples, 6–1
  deleting all files in a directory, 6–18
  deleting directories, 6–18
  displaying job process information, 6–2
  displaying system roots, 6–16
  submitting periodic batch jobs, 6–15
  terminating processes, 6–14
System Management utility (SYSMAN) commands
  IO SET PREFIX, 3–20
  IO SHOW EXCLUDE, 3–20
System messages
  accessing $STATUS values for uninstalled
    messages, 2–12
  new messages, A–1
System parameters
  DBGTK_SCRATCH, 3–17
  GH_EXEC_CODE, 3–17
  GH_EXEC_DATA, 3–17
  GH_RES_CODE, 3–17
  GH_RES_DATA, 3–17
  IMGREG_PAGES, 3–17
  LGI_CALLOUTS, 3–16
  NET_CALLOUTS, 3–16
  PFN_COLOR_COUNT, 3–17
System parameters (cont'd)
  SECURITY_POLICY, 3–16, 3–18
  SHADOWING, 3–18
  SHADOW_MAX_COPY, 3–19
  SHADOW_MBR_TMO, 3–19
  SHADOW_SYS_DISK, 3–19
  SHADOW_SYS_TMO, 3–17
  SHADOW_SYS_UNIT, 3–19
  SHADOW_SYS_WAIT, 3–17
  SYSTEM_CHECK, 3–18
  XQPCTL2, 3–6, 3–18
  XQPCTLD1, 3–18
System security enhancements, 3–9
System services
  $ABORT_TRANS, 4–23
  $ASCUTC, 4–26
  $AUDIT_EVENT, 3–10, 4–26
  $BINUTC, 4–26
  $CHECK_ACCESS, 3–11
  $CHECK_PRIVILEGE, 3–10, 4–26
  $CHKPRO, 3–10
  $CLRCLUEVT, 4–23
  $CREATE_USER_PROFILE, 3–13, 4–26
  $END_TRANS, 4–23
  $FILESCAN, 4–23
  $GETDVI, 4–24
  $GETJPI, 4–12
  $GETQUI, 4–22
  $GETUAI, 4–25
  $GETUTC, 4–27
  $GET_ARITH_EXCEPTION, 4–24
  $GET_SECURITY, 3–13, 4–26
  $IEEE_SET_FP_CONTROL, 4–25
  $NUMUTC, 4–27
  $QIO, 4–12
  $SCHED, 4–25
  $SETCLUEVT, 4–23
  $SETSHLV, 4–12
  $SETUAI, 4–25
  $SET_RESOURCE_DOMAIN, 3–13, 4–26
  $SET_SECURITY, 3–13
  $SNDJBC, 4–22
  $START_TRANS, 4–23
  $SUBSYSTEM, 3–14, 4–26
  $TIMCON, 4–27
  $TSTCLUEVT, 4–23
System startup
  security profiles, 3–12
SYSTEM_CHECK system parameter, 3–18

T

Tape serving, 3–2
Time conversion service, 4–26
Time Differential Factor (TDF), 4–26
TMSCP server, 3–3

TMSCP_LOAD parameter, 3–3
Token Ring
  address mappings, 4–37
  frames, 4–39
  functional address mapping, 4–37
Transport
  LAT, 5–6
TRN controller, 4–36
TULIP controller, 4–36

U

UPGRADE privilege, 3–16
Upgrade procedures
  using the POLYCENTER Software Installation utility, 3–5
/USAGE qualifier
  with the SHOW LICENSE command, 3–4
UTC (Coordinated Universal Time) service
  using to compensate for differing time zones, 4–26

V

VAX C
  See C
Virtual I/O cache, 3–4
VMScluster environments
  security object database, 3–12
  security profiles, 3–12
  tape serving, 3–2
  using multiple queue managers, 3–8
  virtual I/O cache, 3–4
Volume protection, 3–11
Volume shadowing
  on OpenVMS AXP systems, 3–1
VT330 font support, 5–10

W

Write access, 3–13

X

XQPCTL2 system parameter, 3–6, 3–18
XQPCTLD1 system parameter, 3–18
X_floating complex datum, 4–17
X_floating data types, 4–16
X_floating datum, 4–16

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825) and press 2 for technical assistance.

Electronic Orders

If you wish to place an order through your account at the Electronic Store, dial 800-234-1998, using a modem set to 2400- or 9600-baud. You must be using a VT terminal or terminal emulator set at 8 bits, no parity. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825) and ask for an Electronic Store specialist.

Telephone and Direct Mail Orders

From	Call	Write
U.S.A.	DECdirect Phone: 800-DIGITAL (800-344-4825) Fax: (603) 884-5597	Digital Equipment Corporation P.O. Box CS2008 Nashua, NH 03061
Puerto Rico	Phone: (809) 781-0505 Fax: (809) 749-8377	Digital Equipment Caribbean, Inc. 3 Digital Plaza, 1st Street Suite 200 Metro Office Park San Juan, Puerto Rico 00920
Canada	Phone: 800-267-6215 Fax: (613) 592-1946	Digital Equipment of Canada Ltd. 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: DECdirect Sales
International	———	Local Digital subsidiary or approved distributor
Internal Orders[1] (for software documentation)	DTN: 264-3030 (603) 884-3030 Fax: (603) 884-3960	U.S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063-1260
Internal Orders (for hardware documentation)	DTN: 264-3030 (603) 884-3030 Fax: (603) 884-3960	U.S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063-1260

[1]Call to request an Internal Software Order Form (EN–01740–07).

Reader's Comments

OpenVMS AXP Version 6.1
New Features Manual
AA–Q2HNA–TE

Your comments and suggestions help us improve the quality of our publications.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (product works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

_____

_____

What I like best about this manual is _____

_____

_____

What I like least about this manual is _____

_____

_____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

For software manuals, please indicate which version of the software you are using: _____

_____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Phone _____

Do Not Tear - Fold Here and Tape

digital™

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OpenVMS Documentation
110 SPIT BROOK ROAD ZKO3–4/U08
NASHUA, NH 03062–2642

Do Not Tear - Fold Here

Reader's Comments

OpenVMS AXP Version 6.1
New Features Manual
AA–Q2HNA–TE

Your comments and suggestions help us improve the quality of our publications.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (product works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

_____

_____

What I like best about this manual is _____

_____

_____

What I like least about this manual is _____

_____

_____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

For software manuals, please indicate which version of the software you are using: _____

_____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Phone _____

Do Not Tear - Fold Here and Tape

digital™

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OpenVMS Documentation
110 SPIT BROOK ROAD ZKO3–4/U08

NASHUA, NH 03062–2642

Do Not Tear - Fold Here

digital

Printed in U.S.A.